GEODE - Sharing Occupational Data Through the Grid

Proceedings of the UK e-Science All Hands Meeting 2006

Nottingham, UK

18th – 21st September

Editor
Simon J Cox

Published by the National e-Science Centre

ISBN 0-9553988-0-0

Sponsored by
JISC, Microsoft®, ORACLE®, IBM®
OERC, eSI

Premier Sponsor

Joint Information Systems Committee

Gold Sponsor

Silver Sponsor

Bronze Sponsors

Welcome and Introduction

Malcolm Atkinson, e-Science Envoy & Anne Trefethen, Chair of the AHM Steering Committee

This year has seen an important transition for the UK e-Science programme. Many of the first set of pilot projects have completed and we now have clear outputs from the programme. e-Science in the UK had its fifth birthday and moved in to a sustained mode of management with the programme directorate being replaced by an EPSRC programme manager and Malcolm Atkinson as e-Science envoy.

The AHRC began their own e-Science programme and the e-Social science activities have matured, while all of the Research Councils, JISC and other stakeholders continue to develop e-Science and e-Infrastructure.

The theme for this year's UK e-Science All Hands Meeting - *Achievements, Challenges and New Opportunities* - reflected this transition and as you will see in this proceedings reflects the broad range of presentations and discussion held at the meeting. From the results of the completed activities, to the challenges for new areas, from the blue-skies research issues to the industry take up of prototype technologies.

The programme of the meeting included international leaders, workshops, panel sessions, posters and BoFs together with interactive demonstrations at the e-Science Centres' and other booths. In this proceedings we can only capture a slice of the work presented but the papers here indicate the high quality and breadth of the continuing effort in e-Science in the UK.

A special thanks are due to the Steering Committee especially to Ken Brodlie who as Vice Chair will take on the lead for next year's meeting; the Programme Committee, chaired by Paul Watson and Jie Xu, who together with the programme committee members handled the refereeing of the large number of submitted papers; to the team at the National e-Science Centre and EPSRC for organising the event and for the management of the website and the paper submission activity; and to Susan Andrews for her tireless efforts in producing this proceedings.

Finally we would like to acknowledge the support for the meeting from EPSRC, JISC, Microsoft, IBM, ORACLE, Oxford e-Research Centre and the e-Science Institute.

Malcolm Atkinson and Anne Trefethen

AHM 2006 Steering Committee

Anne Trefethen (Chair)	EPSRC
Ken Brodlie (Vice-chair)	University of Leeds
Sheila Anderson	Arts and Humanities Data Service
Susan Andrews	National e-Science Centre
Malcolm Atkinson	National e-Science Centre
Carol Becker	EPSRC
Ann Borda	JISC
John Brooke	University of Manchester
Audrey Canning	DTI
Simon Cox	University of Southampton
Maia Dimitrova	JISC
Jim Fleming	EPSRC
Ned Garnett	NERC
John C Gordon	CCLRC – RAL
Mark Hayes	University of Cambridge
David Ingram	University College London
Anna Kenway	e-Science Institute
Marta Kwiatkowska	University of Birmingham
Andrew Martin	University of Oxford
Alison McCall	National e-Science Centre
Deborah Miller	PPARC
Lelia O'Connell	MRC
Ron Perrott	Belfast e-Science Centre
Adam Staines	BBSRC
David Walker	Cardiff University
Paul Watson	University of Newcastle

Welcome and Introduction

Paul Watson, Chair of the Programme Committee

The AHM 2006 Programme

Dear Colleagues,

This was the first year in which we have required full papers to be submitted for review rather than just abstracts. This change was made as the quality of the All Hands papers has risen steadily since the very first meeting in 2001 and we wanted to ensure that the very best papers were presented at the meeting. We believe that the change has been a success, enabling us to select an exciting set of high quality papers. For this, we are of course very grateful to those who submitted papers, successful or otherwise.

All submissions to AHM 2006 were rigorously reviewed under the supervision of the PC. We received 128 paper regular submissions from which the PC selected 84 papers to be presented. I would like to thank the members of the PC who were able to maintain a high quality of reviewing even though the timescales were much tighter than in previous years. Throughout the whole process, the members were unfailingly helpful, hard-working and supportive. Those other reviewers who provided expert advice on papers in their area were also of great help to us. Overall, I feel that this reflects the importance and warmth with which the AHM is viewed within the e-Science community.

The most oversubscribed part of the conference this year was the Workshop programme for which we received 25 proposals for only 10 slots. It is good to see the growing worldwide interest in the AHM reflected in the UK-Korean and UK-Chinese workshops that we will be hosting. The Birds of a Feather sessions were also popular, with 13 proposals for only 5 places. In the case of both Workshops and BoFs, the PC tried to strike a balance between encouraging the formation of new communities in interesting, emerging areas, and supporting mature communities that have developed over the past years at the AHM.

The overall reviewing process was made straightforward thanks to the work of Susan Andrews of NeSC who not only provided online tools that simplified our work, but also gave me valuable help through the past 8 months, for which I am very grateful.

Finally, I would like to thank the Vice-Chair - Prof Jie Xu – for his constant assistance and good advice.

I hope you find the AHM 2006 programme interesting and enjoyable.

Paul Watson
Chair
AHM 2006 Programme Committee

AHM 2006 Programme Committee

Robert Allan	CCLRC Daresbury
Malcolm Atkinson	National e-Science Centre
Jim Austin	University of York
Dave Berry	National e-Science Centre
Jon Blower	University of Reading
John Brooke	University of Manchester
Kai Cheng	Brunel University
Neil Chue Hong	University of Edinburgh
Peter Clarke	National e-Science Centre
Geoff Coulson	Lancaster University
Simon Cox	University of Southampton
Jon Crowcroft	University of Cambridge
Peter Dew	University of Leeds
Paul Donachy	Queen's University Belfast
Matthew Dovey	Oxford e-Science Centre
Stuart Dunn	King's College London
Alvaro Fernandes	University of Manchester
James Gheel	SAP Research UK
Ian Grimstead	Cardiff University
Mark Hayes	University of Cambridge
Nick Holliman	University of Durham
Yan Huang	Cardiff University
Tom Jackson	University of York
Stephen Jarvis	University of Warwick
Marina Jirotka	University of Oxford
John Kewley	CCLRC Daresbury
Kerstin Kleese Van Dam	CCLRC - Daresbury Laboratory
Ewan Klein	University of Edinburgh
Marta Kwiatkowska	University of Birmingham
Nik Looker	University of Durham
Liz Lyon	UKOLN
Bob Mann	University of Edinburgh
Richard McClatchey	University of the West of England
Steven McGough	Imperial College London
Luc Moreau	University of Southampton
Peter Murray-Rust	University of Cambridge
Dave Newbold	University of Bristol
Steven Newhouse	OMII
Panos Periorellis	University of Newcastle upon Tyne
Ron Perrott	Belfast e-Science Centre
Omer Rana	Cardiff University
Tom Rodden	University of Nottingham
Lakshmi Sastry	CCLRC - RAL
Dimitra Simeonidou	University of Essex
Richard Sinnott	National e-Science Centre - Glasgow
Rob Smith	North East Regional e-Science Centre
Firat Tekiner	University of Manchester
Georgios Theodoropoulos	University of Birmingham

Nigel Thomas	University of Newcastle upon Tyne
David Wallom	Oxford Interdisciplinary e-Research Centre
Paul Watson	University of Newcastle upon Tyne
Kum Won Cho	KISTI
Jie Xu	University of Durham

Other Reviewers

Ali Afzal	Imperial College London
Oluwafemi Ajayi	University of Glasgow
Rob Allan	CCLRC Daresbury Laboratory
John Allen	National e-Science Centre
Stuart Anderson	School of Informatics, University of Edinburgh
Ashiq Anjum	University of the West of England
Ann Apps	University of Manchester
Tobias Blanke	King's College London
Jeremy Bradley	Imperial College London
Christian Brenninkmeijer	University of Manchester
Ken Brodlie	University of Leeds
Daragh Byrne	University of Edinburgh
Mark Calleja	University of Cambridge
Stuart Charters	University of Durham
Dan Chen	University of Birmingham
Zheng Chen	University of Southampton
Philip J. Clark	University of Edinburgh
Jeremy Cohen	Imperial College London
John Colquhoun	North East Regional e-Science Centre
Adrian Conlin	North East Regional e-Science Centre
Greig A Cowan	National e-Science Centre
Clive Davenhall	National e-Science Centre
Matthijs Den Besten	Oxford e-Research Centre
James Farnhill	JISC
Donal Fellows	University of Manchester
Martyn Fletcher	University of York
Brian Foley	University of Warwick
Jon Gibson	University of Manchester
Paul Grace	Lancaster University
Barry Haddow	University of Edinburgh
Michael Hamilton	University of Newcastle upon Tyne
Bruno Harbulot	University of Manchester
LIgang He	University of Cambridge
Cornelia Hedeler	University of Manchester
Jano van Hemert	National e-Science Centre
Hugo Hiden	North East Regional e-Science Centre
Julian Hill	Met Office
Conrad Hughes	University of Edinburgh
Ally Hume	University of Edinburgh
Adrian Jackson	University of Edinburgh
Mark Jessop	University of York
Mike Jones	University of Manchester

Evangelos Kotsovinos	Deutsche Telekom Laboratories
William Lee	Imperial College London
Chun Lei Liu	University of Reading
Peter Maccallum	University of Cambridge
David Meredith	CCLRC Daresbury Laboratory
Simon Miles	University of Southampton
Rob Minson	University of Birmingham
Isi Mitrani	University of Newcastle upon Tyne
Luc Moreau	University of Southampton
Gihan Mudalige	University of Warwick
Arijit Mukherjee	University of Newcastle upon Tyne
Alexsandra Nenadic	University of Manchester
Goran Nenadic	University of Manchester
Christine Niedermeier	University of Newcastle upon Tyne
Mohammed Odeh	University of the West of England
Glenn Patrick	Rutherford Appleton Lab.
Mike Pettipher	University of Manchester
Georgios Pitsilis	University of Newcastle upon Tyne
Duncan Russel	University of Leeds
V. Sastry	Cranfield University
Jennifer Schopf	National e-Science Centre
Thirunavukkarasu Sivaharan	Lancaster University
Tony Solomonides	University of the West of England
Martin Szomszor	University of Southampton
Amine Tafat	University of Cambridge
Victor Tan	University of Southampton
Adel Taweel	University of Birmingham
John Taylor	University of Edinburgh
Aaron Turner	University of York
Rik Tyer	CCLRC Daresbury
Aad van Moorsel	University of Newcastle upon Tyne
David Walker	Cardiff University
John Watt	National e-Science Centre - Glasgow
Jake Wu	University of Newcastle upon Tyne
Xiaobo Yang	CCLRC Daresbury
Steven Young	Oxford University Computing Services
Tianyi Zang	University of Birmingham
Lei Zhao	University of Birmingham

Welcome and Introduction

Simon Cox, editor of the All Hands Proceedings 2006

Welcome to the proceedings of the UK e-Science All Hands meeting 2006. On this CD you will find pdf versions of the papers which were presented in the regular sessions, workshops, and as posters at the conference.

Many thanks to:
- The All Hands Meeting 2006 programme committee for refereeing the papers
- The Team at the National e-Science Centre (NeSC) for their invaluable help
- Our sponsors for these proceedings.

Particular and special thanks to the invaluable Susan Andrews at NeSC, who performed an excellent and thorough job in bringing all aspects of the proceedings together- right from handling the initial submissions to delivering the final CD.

I hope that you will find this a useful resource.

Prof Simon Cox
University of Southampton

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

GOLD Infrastructure for Virtual Organisations

Periorellis P., Cook N., Hiden H., Conlin A., Hamilton M.D.,
Wu J., Bryans J., Gong X., Zhu F., Wright A.

Panayiotis.Periorellis@ncl.ac.uk

May 2006

North East Regional e-Science Center,
School of Computing Science,
Claremont Tower,
University of Newcastle Upon Tyne,
Newcastle, NE1 7RU, UK

Abstract

The principal aim of the GOLD project (Grid-based Information Models to Support the Rapid Innovation of New High Value-Added Chemicals) is to carry out research into enabling technology to support the formation, operation and termination of Virtual Organisations (VOs). This technology has been implemented in the form of a set of middleware components, which address issues such as trust, security, contract monitoring and enforcement, information management and coordination. The paper discusses these elements, presents the services required to implement them and analyzes the architecture and services.The paper follows a top down approach starting with a breif outline on the architectural elements, derived during the requirements engineering phase and demonstrates how these archtiectural elements were mapped onto actual services that were implemented according to SOA principles and related technologies.

1 Introduction to GOLD

GOLD (an EPRSC Pilot project) aims to assist virtual organisations to form, operate and terminate by privinding a set of architectrural components that are dependable while at the same time flexible regarding their adaptation and usage. The project itself went through a thorough process of requirements engineering by investigating the actual needs of potential virtual organisations. It has therefore delivered an architecture that addresses those needs that can be broadly categorised in terms of security, workflow, storage and contract management requirements. The purpose of this paper is to discuss the architectural components while at the same time provide some detail regarding the actual implentation. The paper is structured as follows. Section 2 provides a brief outeline of the architectural elements that comprise the GOLD infrastructure. Section 3 which forms the core of this paper describes the services that were imlpemented to reflect the archtiectural elements discussed in section 2. The paper concludes with section 4. It should be noted here that the majority of requirements were gathered by researching the potential of virtual organisations, forming within the chemical development sector. The team was in close contact with a number of companies from that domain which helped capture the more intricate details of the infrastructure. Nonetheless the infrastructure is not tailored to that specific domain, as it is build as a generic set of services that can be adapted in a flexible manner.

2 Architecture

The GOLD Middleware architecture has primarily been derived through the application of Soft Systems Modelling in addition to a number of interviews that were conducted with companies wihtin the chemical development sector [Periorellis et. al. 2006]. Some of the early findings suggested that the infrastructure needs to be flexible, adaptable and capable of coping with the dynamic characteristics of VOs. In addition, it is undesirable to impose unnecessary constraints on the potential for VO formation by dictating the specifics of the various supporting technologies the entities are required to deploy in order to participate in a VO. Having used these models as a guide we have derived the following architectural that comprise GOLD [Hiden et. al. 2005]. Figure 1 below shows the main elements identified following the analysis of the

SSM model.

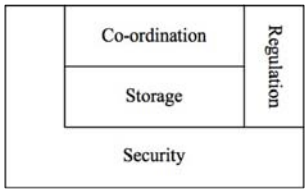


Figure 1 - Architectural elements

To support such a dynamic approach, including the need for late binding and loose coupling to actual implementation technologies, a Service Oriented Architecture (SOA) [Erl 2004] based upon standard Web Services [Skonnard 2002] has been identified as the most suitable means of implementing the GOLD Middleware. Web Services make it possible to use a variety of standards and protocols, and allow the integration of different software applications running on heterogeneous systems. It is important to note, however, that the architectural elements described in the following sections do not necessarily map directly to individual services; rather, they represent high level areas of functionality that require one or more physical services to support them. The security element is paramount encompasses mechanisms for information exchange, access to ressources, user authentication and autorisation. The quantity of information generated in a virtual organisations is significant. This information needs to be stored such that it is available to, and searchable by, correctly authenticated and authorised VO participants. Central to the storage aspect of the GOLD Middleware is the information model describing the topology of the VO and the data and document types that can be exchanged between participating entities. This is a key aspect of the system as it supports the extensibility needed to allow the infrastructure to be tailored to different problem domains. The coordinaation element emphasises the need for planning within a VO. Tasks are coordinated, and will either be performed manually or automatically. Therefore Middleware platforms need to support not only the enactment of pre-determined workflows, but also provide a flexible environment that does not follow a fixed workflow. The Regulation aspect of the architecture aims to ensure that entities who interact within a VO are able to exercise their rights and that, at the same time, they meet their obligations to one another and to any relevant regulatory body.

3 Service Implementations

To support the architectural elements introduced in Section 2, the GOLD project has derived and implemented a number of core services. Figure 2 shows these GOLD services and their relationship to the architectural elements.

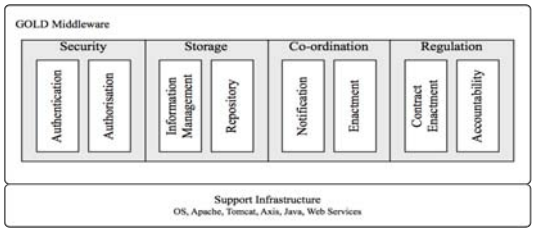


Figure 2 - Web Service oriented infrastructure

This section of the paper describes the technical implementation details of the architecture and describes the set of services currently implemented.

3.1 Security

When granting personnel from an external company access to internal resources it may be necessary to restrict access to those resources. To address this issue, the security element of the GOLD architecture is implemented in the form of authentication and authorisation services which enable members of a VO to define the roles relevant to their project in a central or federated manner. These roles can then be assigned access rights to resources locally, based on the work required to be carried out. In addition these rights can be updated depending upon the stage of a project allowing fine-grained access control.

3.1.1 Authentication

Authentication describes the process of securely establishing and verifying identities of network subjects which may take the form of users, agents, registered services or components. The objectives of the authentication mechanism of the GOLD Middleware are to make sure that only the correct participants enter and operate within the VO and to allow the participants to interact freely (within the range set by the access control policies) with the various services and resources provided by the VO. During the lifetime of a VO its participants will be required to share resources and hence access to those resources will require crossing of organisational boundaries. Clearly, expecting a VO participant to log in several times in order to carry out a task that is part of the same activity is not productive. Several approaches have been proposed, most notably Microsoft Passport [MS Passport 2005] and the Liberty Alliance Group [Liberty 2005]. The GOLD infrastructure supports privacy of a user's own information as long as there is a traceable link between the federated identity and their credentials. For reasons such as data protection and privacy the infrastructure issues a federated identity valid only within the VO. Participants can therefore retain their privacy as the federated identity does not identify the real identity of the participant. This implies that the infrastructure maintains a traceable link between the

federated identity and the real identity of the participant allowing both accountability and privacy to be supported. Secure message exchange between the VO participants is achieved by exchanging security tokens carried within the headers of the exchanged messages. The federation service signs and attaches these tokens to SOAP message headers according to the WS-Security specification [WS-Security 2004]. The specification supports various token structures such as X.509 certificates, Username or XML-based tokens which have been customised, in the GOLD Middleware, to support SAML [OASIS 2004] assertions. The structure of a message carrying a token and the lifecycle of such a token from request to validation is shown in Figure 3.

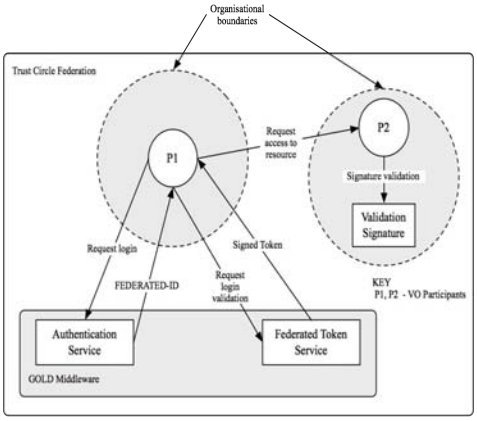


Figure 3 - Security token lifecycle

In the body of the message carries the actual message while the header is structured according to WS-Security specification and carries a security token formatted in one of the standardised styles described earlier. The above description implies that a federation has been established and that there is a direct trust relationship between the federation and the participants. If this direct relationship is not present the above model fails. To alleviate this problem the GOLD Middleware makes use of the WS-Trust [2005] specification that offers a mechanism for managing brokered trust relationships.

3.1.2 Authorisation

In earlier publications [Periorellis et al. 2004] we discussed the advantages and disadvantages of access control models, ranging from active control lists to role and task based systems. We concluded that the dynamic nature of virtual organisations makes it necessary for any VO infrastructure to support a mechanism that deals with dynamic rights activation and de-activation. In dynamic systems such VOs, both the topology of the VO (parties that form it and links between them) and its workflow are subject to change. Therefore static rights assignment prior to a workflow does not capture the eventuality of a party leaving, added or any alterations to the workflow itself. Several authors have elaborated on this issue [Coulouris, G., et al. 1994, Roshan, K., T., et al. 1997, Sandu, R., S., et al. 1996, Thomas R., K., 1997]. In addition, given the sensitivity of the information that may be shared in some VOs, (which raises concerns regarding competitive advantage) parties are not expected to be assigned a single set of rights that would last throughout the duration of a VO. It is more likely that VO participants would agree limited or gradual access to their resources depending on workflow progress. In GOLD we want to be able to restrict role access on resources depending on the execution context. In a virtual organization the execution context can be regarded as the particular project or the goal that all participants have come together to achieve. In a virtual organization there can be many projects with various participants resulting to complicated interrelationships between them. For example, some of them may play the same role in various projects, carrying out the exact same tasks, or have different roles within the same project depending on the performed task. The question we raise is 'Should a role have the same permission and access rights throughout the set of similar or even identical projects and the tasks within those projects?' Our view is that in dynamic access control systems we should separate roles from role instances. To support this, we present the role instances as parameterised roles. Those parameters can be used to express relations to distinguish the role with task-specific or project-specific privileges from the general role. Also some roles may be relational, e.g. originator(document, project) and it may be necessary to enforce separation of duties for such purposes as ensuring that the originator of a document cannot also sign it off. Policy on sign-off could include the possession of certain qualifications in combination with an organisational position, for example the manager can sign off a document provided he/she also has a chemist qualification and is not the originator of that document. Different role instances may require different permissions and indeed additional levels of authorization depending on the project and task in which they are active. To be able to handle such cases, GOLD needs to support adequately fine-grained access control mechanisms. Parameterisation of roles supports both relational roles and fine-grained access control In order to raise the levels of trust in those cases, one needs to make sure that adequately fine grained access control mechanisms are supported. Granularity refers to the level of detail for which one can define access rights. Fine grained permissions are needed for instances of roles as well as instances of objects. For example, a chemist role may be granted access to chemical documents but we do not however wish to grant access to all chemical documents produced by the system. Instead, we want any access permissions granted to the chemist role to be project-specific (e.g., the instance of a particular collaboration) as well as task-specific (e.g., the instance of a particular pre-defined set of activities). So the management of roles and access permissions in

GOLD needs to be integrated with the management and monitoring of dynamic service level agreements or contracts between the participating services. The contracts can capture the expectations from specific tasks, using pre- and post-conditions. Permissions for roles can be activated and de-activated based on the progress of the monitored contracts.. Rights should not be automatically assumed upon role assignment. Instead they should be granted gradually, as the workflow progresses, prohibiting access to parties that may be part of a workflow but are not currently enacting the task for which the access right is relevant. Equally important is that rights may become more restricted with workflow progress, thus the achievement of certain milestones may trigger a permanent change of access rights. Assume a certain project has four phases, Project Evaluation, Route Development, Process Development and Technology Transfer, with a milestone at the end of each phase. Each milestone is an event that will be notified by the Coordination service (see section 3.2 below), to which the Security service subscribes. On receiving this event, the Security service will make appropriate changes to the rights associated with role instances: for example, at the outset of the project four roles - Senior Manager, Financial Analyst, Project manager and Senior Chemist - all have Read and Write permissions on documents within the Marketing Dossier. On achievement of the Route Development milestone, Write permission on these documents is removed from the Financial Analyst and Senior Chemist roles. When the Process Development milestone is reached, write permission on these documents is also removed from the Project Manager role. Notification by the Coordination service to the Security service also supports the dynamic revocation of roles, prohibiting access to parties that are not fulfilling their obligations or who are no longer in the given organisational role. Given these requirements, there are several functionalities which the infrastructure has to support including:

- a common language for expressing authorisation policies that is understood by all participants;
- a protocol for expressing policies and rules that is understood by all participants;
- a protocol for transferring/communicating these policies between VO participants;
- a centralised policy repository;
- a verification component which ensures policy consistency.

Access control policies within the GOLD Middleware are expressed using XACML which is a Web Services standard [OASIS 2003]. XACML has an RBAC profile, which we are extending to provide for parameterised roles .

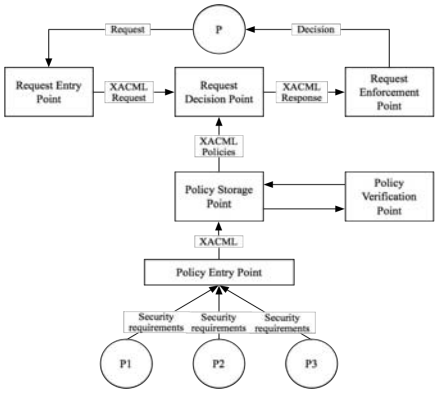


Figure 4 - Access control service architecture

In Figure 4 which illustrates the GOLD access control service, **P** represents a participant and the remainder of the boxes represent the services that have been implemented. VO participants (i.e. **P1**, **P2** and **P3**) express their security requirements using the service interface. This interface provides a user-friendly policy language and also supports consultation of the workflow model, to enable policies to be expressed at the level of specific tasks or subprocesses. These requirements form a set of policies that describe how individual participants need their resources to be protected. The policies for the same project or workflow are placed in temporary storage where a verification service, will validate them for logical inconsistencies. The verification service guarantees that no exceptions will be thrown during VO operations as a result of policy mismatches. Any possible mismatches can be highlighted and compromises negotiated between participants prior to the commencement of VO projects. Assuming that a participant, requests access to a resource, several services coordinate the process of expressing a request and providing a response. Given the wide range of policies that may be required to fully specify the access control requirements within a VO and the fact that there is no single authority governing these policies (the majority of which will stem from participants' requirements on how they want to protect their resources), verification is needed to ensure that there are no logical inconsistencies.

3.2 Coordination

A key aspect of collaborative working is to have some means of ensuring that all interested/involved VO particpants receive coordination messages and notifications. It is also necessary to ensure that tasks are performed at the appropriate time and with the appropriate participants. This leads to a requirement for Notification and Enactment services. VO participants are informed about certain events that take place within the VO through the Notification service. The GOLD infrastructure has adopted the simpler WS-Eventing model which specifies a simple

architecture for subscribing, producing and delivering notification messages. It supports typical subscription operations such as subscribe, unsubscribe and renew, while the delivery mechanism is based on the pull model similar to OMG's CORBA notification model [Bolton 2001]. When the participant subscribes to GOLD's notification service, the service includes information regarding the Subscription Management service in its response. Subsequent operations, such as getting the status of, renewing and unsubscribing, are all directed to the subscription manager. The source (producer) sends both notifications and a message signifying the end of registered subscriptions to the sink (consumer). To provide notification services the GOLD Middleware makes use of NaradaBrokering [Pallickara and Fox 2003] which is a mature, open source, distributed messaging infrastructure. NaradaBrokering can support a number of notification and messaging standards, notably: JMS [JMS 2001], WS-Eventing and WS-Notification. It is, therefore, suitable for intra- and inter-organisational notification. In the context of a VO, a significant advantage of a notification service built on NaradaBrokering is the flexibility of deployment options. The service could be deployed as a stand-alone service at a single site or, alternatively, as a peer-to-peer network of Narada brokers offering a federated notification service. For example, the notification service could be distributed across a set of Trusted Third Parties (TTPs) that support the VO or across the members of the VO itself. In either deployment NaradaBrokering provides scalable, secure and efficient routing of notifications. The Enactment service operates in conjunction with the core GOLD Middleware services to provide support for coordination and workflow. The main components of the service are the workflow editor and the workflow enactment engine, see Figure 5. In addition, there are custom workflow tasks for manipulating VOs and managing documents.

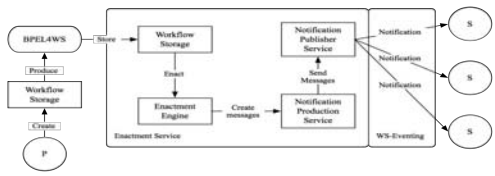


Figure 5 - Enactment service

In Figure 5, participant P describes a workflow using a workflow editor and saves the output in a storage facility. Workflows may be represented graphically using an editor that subsequently creates an XML output document structured according to the BPEL4WS (Business Process Enactment Language for Web Services) specification [BPEL4WS 2002]. The technology provides an XML schema (currently being considered as an OASIS standard) for describing tasks, roles, inputs, outputs, pre and post conditions associated with each task. During workflow enactment, the workflow engine retrieves BPEL4WS documents and executes them, subsequently sending any notifications required to subscribers (S) interested in the state of the coordination activities. GOLD adopted ActiveBPEL [Emmerich et al. 2005] for workflow execution. The Workflow Enactment service responds to requests by initiating workflows to co-ordinate interactions between VO members. For example, the process of approving a chemical safety study may involve passing this study to several approvers, each of which must sign the document before it is considered complete. The enactment engine notifies the participants about events and required activities, depending on the topics participants have registered for. For situations where it is not necessary to co-ordinate the detailed activities of VO participants, the GOLD Middleware can be used to provide an abstraction with a lower degree of integration, for instance a shared space that contains projects that are divided into discrete tasks. Each of these tasks can contain a set of data, which can be accessed and edited by participants with the correct security credentials. These documents can include any of the information types supported within the VO, including the results of any enacted workflows, and any workflows that are still on-going.

3.3 Regulation

Regulation helps govern interactions between parties, ensuring that participants' rights (in terms of the resources they are allowed to access) are properly granted and that obligations are properly dispatched (such as making resources available to others). This is achieved by the use of contracts and contract enforcement mechanisms as well as monitoring mechanisms for auditing and accountability.

3.3.1 Contract Enactment

Each member of a VO requires that their interests are protected, specifically:

- that other members comply with contracts governing the VO;
- that their own legitimate actions (such as delivery of work, commission of service) are recognised;
- that other members are accountable for their actions.

To support this, the GOLD Middleware records all activities to monitor for compliance with the regulatory regime. Furthermore, critical interactions between VO participants should be non-repudiable (no party should be able to deny their participation) and the auditing and monitoring functions must be fair (misbehaviour should not disadvantage well-behaved parties). For example, a business contract governing the scenario described in Section will specify sequencing constraints on the production of documents such as the requirements, recipe, thermal analysis and scale-up analysis. It will also require that the safety of the scaled-up process is assured, hence

the requirement that the the Scale-up company employ the Thermal Safety company to provide an analysis and validation of the potential exotherm that was identified during the scale-up studies. In a complex natural language contract of the form typically negotiated between business partners, there may in fact be ambiguities that, given the sequencing constraints, could lead to deadlock during the chemical process development project. There is, therefore, a need to produce an electronic version of the contract that can be model-checked for correctness properties (e.g. safety and liveness). Having developed a contract that is free of ambiguities, it should then be possible to derive an executable version to regulate an interaction at run-time. That is, the business messages exchanged between participants during the process development should be validated with respect to the executable contract to ensure that they comply with contractual constraints. To hold participants to account, and to be able to resolve subsequent disputes, these message exchanges should be audited. In the scenario described in Section , the Scale-up company sends the Supplier company the scale-up model. The delivery of this document should be validated against the contract to ensure any pre-requisite obligations have been fulfilled. To safeguard their interests, the Supplier company will require evidence that the model originated at the Scale-up company. Similarly, the Scale-up company will require evidence that they fulfilled their obligation to deliver the model to the Supplier company.

Given these requirements, we identify two main aspects to contract enactment and accountability:

- high level mechanisms to encode business contracts so that they are amenable to integrity and consistency checking and in order to derive an executable form of contract;
- a middleware infrastructure that can monitor a given interaction for conformance with the executable contract - ensuring accountability and acknowledgement.

To address the first aspect, Section 3.3.1.1 provides a summary of work on the derivation of electronic contracts and deployment models for contract mediated interaction. This work appears in Molina et al. [2005], which also presents related work. The GOLD project extends this work to enact business contracts using infrastructure for accountability and non-repudiation as an enforcement mechanism. Section 3.3.2 presents this infrastructure and shows how it addresses the second aspect identified above. This paper is concerned with monitoring and enforcement of business operation clauses, of equal importance is the monitoring of the levels of Quality of Service (QoS) offered within a VO. This concerns the collection of statistical metrics about the performance of a service to evaluate whether a provider complies with the QoS that the consumer expects. Molina et al. [2004] examine this aspect of regulation and related work.

3.3.1.1 Contract-mediated interaction

The rules in a conventional, paper-based contract express what operations business partners are:

- permitted to perform if deemed necessary to fulfill the contract;
- obliged to perform to ensure contract fulfillment;
- prohibited from performing as these actions would be illegal, unfair or detrimental to the other partners.

In addition, rules may stipulate when and in what order the operations are to be executed. To form and have automatic management of partnerships within a VO, electronic representations of contracts must be used to mediate the rights and obligations that each member promises to honour. In the worst case, violations of agreed interactions are detected and all interested parties are notified. In order to support this, the original natural language contract that is in place to govern interactions between participants has to undergo a conversion process from its original format into an executable contract (x-contract) that works as a mediator of the business conversations. This conversion process involves the creation, with the help of a formal notation, of one or more computational models of the contract with different levels of details. To achieve these objectives, the Promela modeling language [Holzmann 1991] is used to represent all the basic parameters that typical business contracts comprise, such as permissions, obligations, prohibitions, actors (agents), time constraints, and message type checking. The Promela representation can be validated with the help of the accompanying Spin model-checker tool [Holzmann 2004]. For example, model-checking the Promela representation can improve the original natural language contract by removing various forms of inconsistency as discussed in Solaiman et al. [2003]. This implementation-neutral representation can be refined to include technical details such as acknowledgment and synchronisation messages. The details will vary depending on specific implementation techniques and standards that are adopted. This implementation specific representation can then be used for run-time monitoring. Conceptually, an x-contract is placed between VO members to regulate their business interactions. In terms of the interaction model, the x-contract may be reactive or proactive. A reactive x-contract intercepts business messages, validates the messages and rejects invalid messages. A proactive x-contract drives the cross-organisational business process by inviting VO members to send legal messages of the right type, in correct sequence, at the correct time etc. Deployment can be either centralised or distributed. This leads to four deployment models:

- *Reactive central* - where all messages are intercepted by a centralised x-contract (at a TTP, for example) that is responsible for forwarding just the legal messages to their intended destination.

- *Proactive central* - where a centralised x-contract coordinates the business process on behalf of VO members and triggers the exchange of legal messages.
- *Reactive distributed* - where the x-contract is split into separate components that can be used to validate just those messages sent to an individual VO member and to reject illegal messages sent to that member.
- *Proactive distributed* - a distributed version of proactive central that coordinates the legal participation of each member in the business process.

Distributed deployments face the difficult challenge of keeping contract state information synchronised at both ends. For example, a valid message forwarded by the buyer's x-contract could be dropped at the seller's end because intervening communication delays render the message untimely (and therefore invalid) at the seller side. State synchronisation is necessary to ensure that both the parties agree to treat the message as either valid or invalid. One approach that uses a non-repudiable state synchronisation protocol [Cook et al. 2002] is described in Molina et al. [2003]. The GOLD Middleware used to invoke validation with respect to a contract at runtime is discussed below in Section 3.3.2.

3.3.2 Accountability

This section focuses on accountability for the delivery of a single business message. However, this validated and non-repudiable message delivery can then be used as a building block for contract monitoring and enforcement of the kind envisaged in Section 3.3.1.

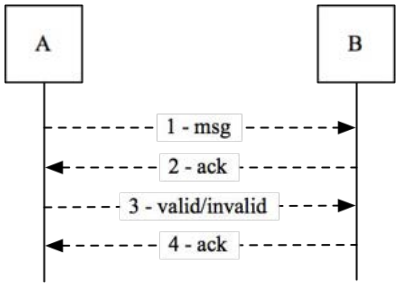


Figure 6 - Business message delivery with acknowledgements

Figure 6 shows the delivery of a business message and associated acknowledgements. Typically, for each business message, there should be an immediate acknowledgement of receipt indicating successful delivery of the message. Eventually, a second acknowledgement indicates whether the original business message is valid (or invalid) in the context of the given interaction. Finally, the validation message is acknowledged in return. Validation of the original business message (performed at **B**) can be arbitrarily complex. For example, it may simply involve verification that a message is syntactically valid and in the correct sequence with respect to a contract. Alternatively, a message may require validation with respect to more complex contractual conditions or with respect to local application state. Triggering validation at the level of business message delivery has the potential to allow specialisation of an application to meet the constraints of different regulatory regimes. Web Services are increasingly used to enable B2B interactions of this kind. However, there is currently no support to make the exchange of a set of business messages (and their associated acknowledgements) both fair and non-repudiable. A flexible framework for fair, non-repudiable message delivery has therefore been developed. The Web Services implementation of this framework comprises a set of services that are invoked at the middleware level and so enable the Web Services developer to concentrate on business functions. The GOLD Middleware renders the exchange of business messages fair and non-repudiable. Arbitrarily complex, application-level validation is supported through the registration of message validators. The framework is sufficiently flexible to adapt to different application requirements and, in particular, to execute different non-repudiation protocols to meet those requirements.

3.3.2.1 Basic concepts

Non-repudiation is the inability to deny an action or event. In the context of distributed systems, non-repudiation is applied to the sending and receiving of messages. For example, for the delivery of a message from A to B the following types of non-repudiation may be required:

- *NRO* - B may require Non-Repudiation of Origin of the message, i.e. irrefutable evidence that the message originated at A;
- *NRR* - A may require Non-Repudiation of Receipt of the message, i.e. irrefutable evidence that B received the message.

Non-repudiation is usually achieved using public key cryptography. If A signs a message with their private key, B can confirm the origin of the message by verifying the signature using A's public key, and vice versa. An additional requirement is that at the end of the interaction no well-behaved party is disadvantaged. For example, consider the selective receipt problem where a sender provides NRO but the recipient does not provide the corresponding NRR. This problem is addressed by the fair exchange of items where fairness is the property that all parties obtain their expected items or no party receives any useful information about the items to be exchanged [Markowitch et al. 2002]. Kremer et al. [2002] provide a survey of protocols to achieve fair, non-repudiable exchange of messages. The following discussion is based on the use of an in-line TTP to support the exchange. However, our execution framework is not restricted to this class of protocol.

3.3.2.2 Overview of approach

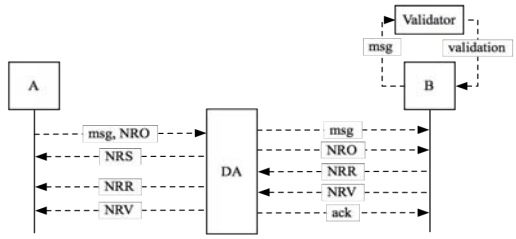


Figure 7 - Executing a business interaction through a delivery agent

Figure 7 introduces a Delivery Agent (DA), or inline TTP, to the interaction shown in Figure 6. Four types of evidence are generated:

- *NRO* - Non-Repudiation of origin that *msg* originated at A;
- *NRS* - Non-Repudiation of submission to the DA of *msg* and NRO;
- *NRR* - Non-Repudiation of receipt of *msg* by B;
- *NRV* - Non-Repudiation of validation, valid or otherwise, as determined by validation of *msg* by B.

A starts an exchange by sending a message, with proof of origin, to the DA. This is the equivalent of Message 1 in Figure 6 with the NRO appended. The DA exchanges *msg* and NRO for NRR with **B** (before application-level validation of *msg*). Then the DA provides NRR to **A** equivalent to Message 2 in Figure 6. Subsequently, **B** performs application-level validation of *msg* (as in Message 3 of Figure 6 and provides NRV to the DA. The DA, in turn, provides NRV to **A** and provides acknowledgement of NRV to **B**. The exact sequence of the exchange will be dictated by the actual protocol used and should not be inferred from Figure 7.

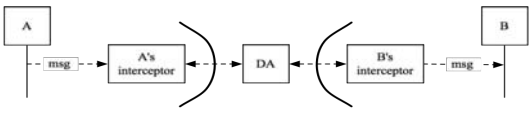


Figure 8 - Interceptor approach

As shown in Figure 8, our approach is to deploy interceptors that act on behalf of the end users in an interaction. An interceptor has two main functions:

- to protect the interests of the party on whose behalf it acts by executing appropriate protocols and accessing appropriate services, including TTP services;
- to abstract away the detail of the mechanisms used to render an interaction safe and reliable for its end user.

In this case, the mechanism used communicates through the DA. It is the responsibility of the DA to ensure fairness and liveness for well-behaved parties in interactions that the DA supports. The introduction of interceptors means, that as far as possible, **A** and **B** are free to concentrate on application level concerns while their interaction is rendered fair and non-repudiable.

3.4 Storage

The storage element addresses the need to store, manage and access information. In addition there is a requirement to be able to determine how a piece of information was derived. The Information Management and Repository services meet this need by providing configurable information storage and logging/auditing functionality.VOs must control and manage the exchange of information between the participants, and the role of the Information Management service in the GOLD Middleware is to support this exchange in three ways:

- to ensure a common structure and meaning for information shared across the VO;
- to provide information services and tools to support the controlled exchange of information according to the policies and contracts that are in place within the VO;
- to extract value from the information stored during the lifetime of a VO.

To support the information management requirements of VOs the GOLD Middleware provides an Information Model that defines the structure and meaning of information shared by its participants. This model can be divided into three categories:

- *Generic* - represents information that is required by all VOs. This includes descriptions of the VO structure, the participants, the tasks being performed, security policies etc. The services that make up the generic GOLD VO infrastructure (i.e. those comprising the security, coordination and regulation architectural elements) all exchange information defined in this category of the information model.
- *Domain specific* - within a particular domain, there are types of information that are generic across a broad range of VOs.
- *Application specific* - information in this category represents specialist information describing a particular domain.

This information model is based on the myGrid information model [Sharman et al. 2004], which was designed to support VOs in the bioinfomatics domain.

4 Conclusions

GOLD middleware offers a set of services that can be used to assist in the formation, operation and termination of virtual organisations. The aim of the project and the proposed architecture is to offer VO developers the flexibility to configure the VO

according to their requirements without imposing too many constraints or imposing what and how it should be done. In this limited space we touched on 4 fundamental architectural elements and discussed in turn how they could be implemented. Adhering to certain principles regarding privacy and trust we devised a security policy for authorisation and authentication that is based primarily on current WS standards. Virtual organisations bring together a number of independent entities with the aim to collaborate in achieving a common goal. This creates the need for some form of coordination regarding the message exchanges between those entities. Coordination therefore is a key aspect of collaborative working. Participants have to remain informed of certain events and It is also necessary to ensure their obligations are dispatched at the appropriate time. The paper showed how regulation helps govern interactions between parties, to ensure that obligations are properly dispatched and rights are properly granted by the use of contracts and contract enforcement mechanisms as well as monitoring mechanisms for auditing and accountability. Finally shed some light was on the issue of storage and information management and as such several broad requirements and implementation strategies were discussed.

References

BOLTON, F. 2001, Pure CORBA, SAMS, ISBN 0672318121

BPEL4WS. 2002. *BPEL4WS V1.1 specification*. ftp://www6.software.ibm.com/software/developer/library/ws-bpel1.pdf.

CHADWICK, D. and OTENKO, A. 2003. The PERMIS X.509 role based privilege management infrastructure. *Future Generation Computer Systems*, 19 , 2, 277-289.

CHECKLAND, P. B. and SCHOLES, J. 1990. *Soft Systems Methodology in Action*, John Wiley and Sons, Chichester.

CONLIN, A.K., ENGLISH, P.J., HIDEN, H.G., MORRIS, A.J, SMITH, R. and WRIGHT, A.R. 2005. A Computer Architecture to Support the Operation of Virtual Organisations for the Chemical Development Lifecycle. In Proceedings of European Symposium on Computer Aided Process Engineering, (ESCAPE 15), 1597-1602.

COOK, N., SHRIVASTAVA, S. and WHEATER, S. 2002. Distributed Object Middleware to Support Dependable Information Sharing between Organisations. In Proceedings of IEEE Int. Conf. on Dependable Systems and Networks (DSN), Washington DC, USA.

COULOURIS, G., and DOLLIMORE, J. 1994. Security Requirements for Cooperative Work: A Model and its System Implications. In Proceedings of Workshop on ACM SIGOPS European Workshop: Matching Operating Systems to Application Needs. Wadern, Germany.

DEMCHENKO, Y., 2004. Virtual organisations in computer grids and identity management. Information Security Technical Report, 9, 1, 59-76.

EMMERICH, W., BUTCHART, L., CHEN, L., WASSERMANN, B., and PRICE, S. L., 2005. Grid Service Orchestration using the Business Process Execution Language (BPEL). UCL-CS. Research Note RN/05/07. Gower St, London WC1E 6BT, UK.

ERL, T. 2004. Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Prentice Hall PTR, 0131428985

FISLER, K., KRISHNAMURTHI, S., MEYEROVICH, L.A. and TSCHANTZ, M.C. 2005. Verification and Change-Impact Analysis of Access-Control Policies. In Proceedings of the 27th International Conference on Software Engineering, 21, 196-205, St. Louis, MO, USA.

GREENBERG, M.M., MARKS, C., MEYEROVICH, L.A. and TSCHANTZ, M.C. 2005. The Soundness and Completeness of Margrave with Respect to a Subset of XACML. Technical Report CS-05-05, Department of Computer Science, Brown University.

HOLZMANN, G.J. 1991. *Design and Validation of Computer Protocols*, Prentice Hall.

HOLZMANN, G.J. 2004. T*he SPIN Model Checker, Primer and Reference Manual*, Prentice Hall.

JMS. 2001. *JMS: Java Messaging Service Specification (JMS)*, http://java.sun.com/products/jms/docs.html.

KREMER, S., MARKOWITCH, O. and ZHOU, J. 2002. An Intensive Survey of Fair Non-repudiation Protocols, Computer Communications, 25, 1601–1621.

KRISHNA, A., TAN, V., LAWLEY, R., MILES, S. and MOREAU, L. 2003. The myGrid Notification Service. In Proceedings of UK OST e-Science All Hands Meeting (AHM'03), Nottingham, UK.

LIBERTY. 2005. *Specification documentation of Liberty Alliance Project.* https://www.projectliberty.org/resources/specifications.php.

MARKOWITCH, O., GOLLMANN, D. and KREMER, S. 2002. On Fairness in Exchange Protocols. In Proceedings of 5th International Conference on Information Security and Cryptology (ISISC 2002), Springer LNCS 2587.

MOLINA-JIMENEZ, C., SHRIVASTAVA, S., CROWCROFT, J. and GEVROS, P. 2004. On the Monitoring of Contractual Service Level Agreements. In Proceedings of IEEE International Conference on E-Commerce (CEC), 1st International Workshop on Electronic Contracting (WEC), San Diego.

MOLINA-JIMENEZ, C., SHRIVASTAVA, S., SOLAIMAN, E. and WARNE, J. 2003. Contract Representation for Run-time Monitoring and Enforcement. In Proceedings of IEEE International Conference On E-Commerce (CEC), Newport Beach, USA.

MOLINA-JIMENEZ, C., SHRIVASTAVA, S., SOLAIMAN, E. and WARNE, J. 2005. A Method for Specifying Contract Mediated Interactions. In Proceedings of 9th IEEE International Enterprise Computing Conference (EDOC), Enschede, Netherlands.

MORGAN, R.L., CANTOR, S., CARMODY, S.,

HOEHN, W. and KLINGENSTEIN K. 2004. Federated Security: The Shibboleth Approach. Educause Quarterly, 27, 4.

MS Passport 2005, http://www.passport.net

NORFOLK, D. 1995. The Virtual Enterprise, Information Age, November, 32-39.

OASIS. 2003. eXtensible Access Control Markup Language (XACML) Version 1.0. OASIS Standard, http://www.oasis-open.org/committees/xacml.

OASIS. 2004. *Security Assertion Markup Language (SAML) v2.0*. http://www.oasis-open.org/committees/security.

PALLICKARA, S. and FOX, G. 2003. NaradaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to- Peer Grids. In Proceedings of ACM/IFIP/USENIX Int. Middleware Conf., Rio de Janeiro, Brazil.

PERIORELLIS, P., TOWNSON, C.J.W., DUNNING-LEWIS, P. and ENGLISH, P.J. 2004. Draft GOLD Requirements Document v1.0, *Technical Report 854*, School of Computing Science, University of Newcastle upon Tyne.

PERRIN, T., ANDIVAHIS, D., CRUELLAS, J.C., HIRSCH, F., KASSELMAN, P., KUEHNE, A., MESSING, J., MOSES, T., POPE, N., SALZ, R. and SHALLOW, E. 2003. Digital Signature Service Core Protocols and Elements. OASIS Committee Working Draft, http://www.oasis-open.org/committees/dss.

ROBINSON, P., COOK, N. and SHRIVASTAVA, S. 2005. Implementing Fair Non-repudiable Interactions with Web Services. In Proceedings of 9th IEEE International Enterprise Computing Conference (EDOC), Enschede, Netherlands.

ROSHAN, K.T. and SANDHU, R.S., 1997. Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-Oriented Authorization Management. In Proceedings of IFIP TC11 WG11.3 Eleventh International Conference on Database Securty XI: Status and Prospects.

SANDHU, R.S., COYNE, E.J., FEINSTEIN, H.L. and YOUMAN, C.E. 1996. Role-Based Access Control Models, IEEE Computer, 29, 38-47.

SHARMAN, N., ALPDEMIR, N., FERRIS, J., GREENWOOD, M., LI, P. and WROE, C. 2004. The myGrid Information Model. In Proceedings of the UK e-Science All Hands Meeting 2004, 1 September.

SKONNARD, A. 2002. The XML Files: The birth of Web Services, MSDN Magazine, 17, 10.

SMITH R. 2005. Defining Virtual Organisations. *Technical Report 965*, School of Computing Science, University of Newcastle upon Tyne.

SOLAIMAN, E., MOLINA-JIMENEZ, C. and SHRIVASTAVA, S. 2003. Model Checking Correctness Properties of Electronic Contracts. In Proceedings of International Conference on Service Oriented Computing (ICSOC), Springer LNCS 2910, Trento, Italy.

THOMAS, R. K. 1997. Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments. In Proceedings of Second ACM Workshop on Role-based Access Control, Fairfax, Virginia, United States.

WS-EVENTING. 2004. Web Services Eventing Specification (WS-Eventing), http://www-128.ibm.com/developerworks/webservices/library/specification/ws-eventing/

WS-NOTIFICATION. 2005. Web Services Notification Draft Specifications (WS-Notification), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn

WS-RELIABLEMESSAGING. 2005. *Web Services Reliable Messaging Protocol (WS-ReliableMessaging)*. http://msdn.microsoft.com/ws/2005/02/ws-reliablemessaging/.

WS-RELIABILITY. 2004. Web Services Reliable Messaging TC WS-Reliability 1.1. OASIS Committee Working Draft, http://www.oasis-open.org/committees/wsrm/.

WS-SECURITY. 2004. Web Services Security: SOAP Message Security 1.0 (WS-Security 2004). OASIS Standard 200401, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf.

WS-TRUST. 2004. *Web Services Trust Language (WS-Trust)*. http://msdn.microsoft.com/ws/2004/04/ws-trust/.

WU, J. and PERIORELLIS, P. 2005a. Authorization-Authentication Using XACML and SAML. *Technical Report 907*, School of Computing Science, University of Newcastle upon Tyne.

WU, J. and PERIORELLIS, P. 2005b. Evaluation of autorisation-Authentication tools. *Technical Report 935*, School of Computing Science, University of Newcastle upon Tyne.

XKMS. 2005. XML Key Management Specification (XKMS 2.0). W3C Recommendation, http://www.w3.org/TR/xkms2/.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Legacy Code Support for Commercial Production Grids

Gabor Terstyanszky[1], Tamas Kiss[1], Peter Kacsuk[1,2], Thierry Delaitre[1], Gabor Kecskemeti[2], Stephen Winter[1]

[1]Centre for Parallel Computing, University of Westminster
115 New Cavendish Street, London, W1W 6UW
e-mail: gemlca-discuss@cpc.wmin.ac.uk

[2]MTA SZTAKI, 1111 Kende utca 13
Budapest, Hungary

Abstract

Currently several production Grids offer their resources for academic communities. Access to these Grids is free but restricted to academic users. As so, the Grids offer only basic quality of service guarantees and minimal user support. This incorporates Grid portals without workflow editing and execution capabilities, brokering with no QoS and SLA management, security solutions without privacy and trust management. Today's Grids do not provide any kind of support for running legacy code applications, and do only very basic accounting without any billing functionalities. This academic experimental phase should be followed by opening up the Grid to non-academic communities, such as business and industry. The widening of the Grid user community defines additional requirements. This paper discusses how the GEMLCA legacy code architecture is extended in order to fulfil the requirements of commercial production Grids. The aim of our research is to facilitate the seamless integration of GEMLCA into future commercial Grids. GEMLCA, in this way, could serve as a reference implementation for any third party utility service that can be added to production Grids in order to improve their usability.

1. Introduction

Grid computing has the potential to move from academic and research communities towards commercial applications and become a major force in business and industry. Some companies made huge investments in compute, storage and other resources in the 1990s, which sit idle most of the time. Some other companies do not have the resources they need in order to solve their business problems. Grids offer solutions for both categories of companies by integrating their heterogeneous resources and providing on-demand computing power. Grid computing connects distributed resources of companies, harnesses their collective resources, and manages them as a single resource.

To make Grid computing available for business and industry, commercial production Grids have to be created that provide stable, production quality infrastructures based on Grid economics. These commercial production Grids should utilise the experiences of existing academic production Grids and should be built on available Grid software technologies. However, several aspects, like virtual organisation management (privacy, security and trust), performance-oriented service management (brokers and information systems) and user-friendly problem solving environments (collaborative Grid portals) have to be added to or significantly improved in current academic Grids in order to fulfil business requirements. Today's academic production Grids are not based on Grid economics, they do not provide reliability and robustness required for business and industry applications, and they do not handle Quality of Service (QoS) and Service Level Agreements (SLA). Production Grids can efficiently serve the research community even without these attributes which, on the other hand, are crucial for a wider industrial take-up of Grid technology.

GEMLCA [1], Grid Execution Management for Legacy Code Applications is a generic solution in order to deploy legacy applications as Grid services without modifying or even accessing the original source code. GEMLCA has been successfully used to Grid-enable several applications [2] [3], and has been offered as a service on the UK National Grid Service (NGS) [4] and the WestFocus Grid [5]. In order to support these academic Grids at production level additional features, like dynamic account management and GEMLCA service monitoring had to be added to the original architecture. However, in order to

integrate GEMLCA with future commercial Grids even more additional features, like accounting and charging based on SLAs, are required.

This paper explores some of the enhancements that have already been added to GEMLCA, or currently under specification and development aiming to fulfil the requirements of both academic and commercial production Grids.

2. Production Grids and Legacy Code Support

There are several production Grid systems, like the TeraGrid [6] and the Open Science Grid (OSG) [7] in the US, or the EGEE Grid [8] and the UK National Grid Service [4] in Europe, that already provide reliable production quality access to computational and data resources for the academic community. All these Grids were set up as resource-oriented Grids based on Globus Toolkit version 2 (GT2) [9]. All these consider moving towards a service-oriented architecture using either gLite [10] or GT4 [11]. They also consider providing service not only for the academic communities but for business and industry too. These objectives require significant enhancement of these Grid infrastructures and also the incorporation of additional user support services.

Enabling the seamless migration of legacy applications onto the new Grid platforms is one of the most important of these user support services. There is a vast legacy of applications solving scientific problems or supporting business critical functionalities which should be ported onto the Grid with the least possible effort and cost. The only solution that supports this functionality at production level and has already been integrated (like in case of the UK NGS), or will be integrated in the near future (like the OSG) to productions Grids, is GEMLCA [1], developed by the University of Westminster. It "gridifies" legacy code applications, i.e. converts these applications into Grid services. The novelty of the GEMLCA concept, compared to similar solutions like in [12], [13] and [14] is that it requires minimal effort from both system administrators and end-users of the Grid providing a high-level user-friendly environment to deploy and execute legacy codes on service-oriented Grids. The deployment of a new legacy code service with GEMLCA means to expose the functionalities of this legacy application as a Grid service that requires the description of the program's execution environment and input/output parameters in an XML-based Legacy Code Interface Description (LCID) file. This file is used by the GEMLCA Resource layer to handle the legacy application as a Grid service.

GEMLCA provides the capability to convert legacy codes into Grid services. However, end-users still require a user-friendly Web interface (portal) to access the GEMLCA functionalities: to deploy, execute and retrieve results from legacy applications. Instead of developing a new custom Grid portal, GEMLCA was integrated with the workflow-oriented P-GRADE Grid portal [15] extending its functionalities with new portlets. The P-GRADE portal enables the graphical development of workflows consisting of various types of executable components (sequential, MPI or PVM

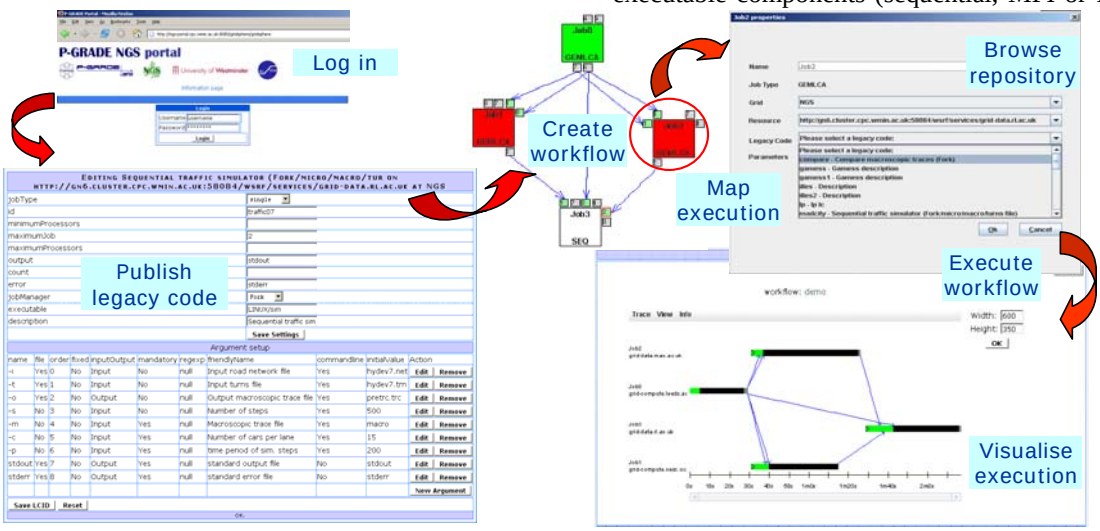


Figure 1: Functionalities of the P-GRADE/GEMLCA Portal

programs), execution of these workflows in Globus-based Grids relying on user credentials, and finally the analysis of the correctness and performance of applications by the built-in visualisation facilities. The portal is based on the GridSphere [16] portal framework and the workflow manager subsystem is currently implemented on top of Condor DAGMan [17].

Following the integration of GEMLCA and the P-GRADE portal, end-users can easily construct workflow applications also including legacy code services running on different GEMLCA Grid resources. The workflow manager of the portal contacts the selected GEMLCA resources, passes them the actual parameter values of the legacy code, and then it is the task of the GEMLCA Resource to execute the legacy code with these parameter values. The other important task of the GEMLCA Resource is to deliver the results of the legacy code service back to the portal. The P-GRADE portal was also extended with the GEMLCA Administration portlet. This portlet manages the XML-based Legacy Code Interface Description (LCID) files, which describe the execution environment and the parameter set of legacy applications. The portlet creates automatically the LCID files and uploads them to the appropriate directory of the GEMLCA resource.

The functionalities of the integrated P-GRADE/GEMLCA portal are represented on Fig. 1. It shows the GEMLCA Administration portlet to publish a new legacy code, the workflow editor to graphically create a new workflow, define its properties and map its execution to Grid resources, and finally to visualize the execution of the workflow. For more detailed description of GEMLCA and the P-GRADE portal please refer to [1] and [15], respectively.

In order to offer GEMLCA as a production level service through the P-GRADE portal for the UK NGS, some further developments of the architecture were necessary: GEMLCA is extended with dynamic user management and service monitoring capabilities. Work is also in progress to define and implement an accounting and charging service that administers GEMLCA resource usage and provides billing information. This extension is not necessary in case of the NGS but crucial for commercial Grid applications. The next chapter describes these current research efforts in detail.

3. GEMLCA Extensions to Support Production Grids

3.1 Dynamic User Management in GEMLCA

As GEMLCA uses local job managers, like Condor or PBS, to execute legacy code jobs through GT4 job submission, it requires a secure run-time environment. In order to achieve this Grid certificates have to be mapped to local user accounts. However, in case of a production Grid it is not scalable to create user accounts and do the mapping manually whenever a new user is added. In a production Grid environment users' Grid credentials have to be mapped dynamically to local user accounts in a completely user-transparent way. Current GT2-based production Grids all tackles this problem. However, there are only limited solutions for GT4-based Grids. GEMLCA is capable to submit jobs to and work seamlessly together with GT2-based Grids like the UK NGS but its internal implementation is based on GT4. Because of this, dynamic user management of GT4-based GEMLCA resources had to be resolved. This section describes a unique architecture how dynamic mapping can be integrated into GT4-based Grid services like GEMLCA, allowing the seamless integration of these services into production Grids.

To provide a scalable user management GEMLCA was integrated with the Workspace Management Service [18] using its identity mapping and dynamic account pooling features. The first feature maps Grid certificates into local accounts at run-time without manual intervention, while the second feature provides local accounts on demand. The previous GEMLCA security implementation [19] used the grid-mapfile to check authorization of Grid service requests. To avoid manual interaction, authorisation of GEMLCA services, such as *GLCList*, *GLCProcess* and *GLCAdmin* [1], was adapted to the WMS Authorisation and Mapping Callout to make GEMLCA scalable. To get a workspace, *GLCProcess* issues a request on the user's behalf to the WMS to create and lease a workspace for the user. Leased workspaces allow Grid users to access GEMLCA resources and allow them to make subsequent service requests. The *GLCProcess* can extend the lease as required, for example until the service completes, during the execution of the legacy codes using a thread associated with the *GLCProcess* environment.

The GEMLCA lifecycle with WMS incorporates the following steps (Figure 2) (Please note that the

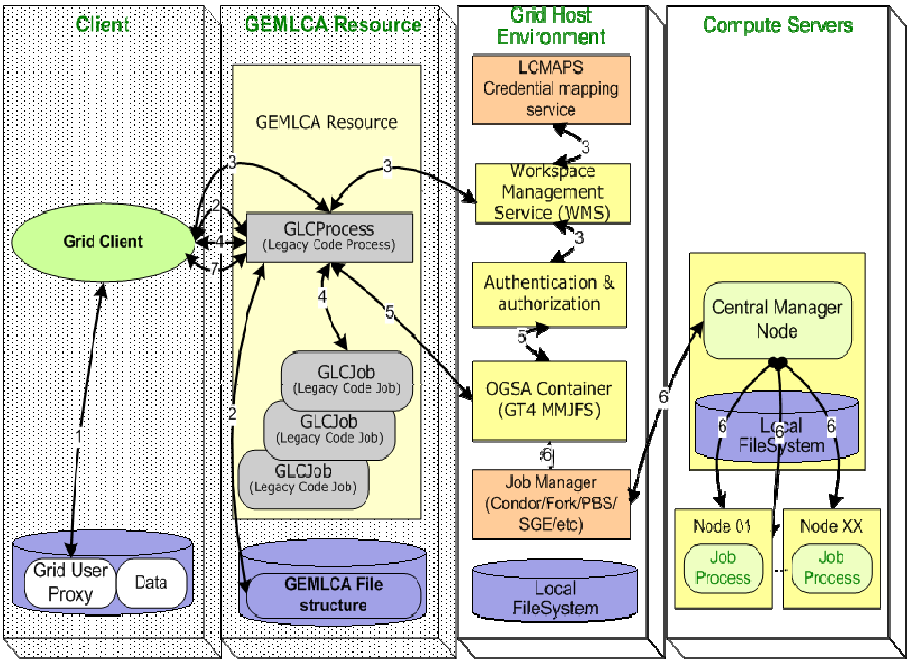


Figure 2: GEMLCA Lifecycle with Workspace Management

original GEMLCA lifecycle and its detailed description are available in [1]. Here only the changes required by dynamic account management are presented):

1. The user signs its security certificates in order to create a Grid user proxy
2. A Grid client creates a restricted Grid Legacy Code Process (*GLCProcess)* instance with no defined workspace, using the GEMLCA file structure.
3. The *GLCProcess* instance forwards the Grid user credential to the Workspace Management Service (WMS) that checks whether a workspace has been previously assigned to the user. If has not been, WMS converts global user identity to local one using the LCMAPS and selects a workspace from the workspace pool assigning it to the user with a lease before creating a *GLCProcess* environment. If the lease is about to expire *GLCProcess* contacts the WMS in order to extend it. As these steps are programmed within the *GLCProcess*, the dynamic creation and mapping of workspaces are totally transparent from the users' point of view.
4. Having a workspace allocated for the user, the Grid client sets and uploads the input parameters needed by the legacy code program exposed by the *GLCProcess*, deploys a *GLCJob* using the resource specification format of the Grid middleware (a Resource Specification Language (RSL) file in case of Globus), and creates a multi-user environment to handle input and output data.
5. The Grid user credential is delegated by the *GLCProcess* from the client to the underlying Grid Host Environment for the allocation of resources. For example, in case of a Globus-based implementation the resource allocation is the task of the Master Managed Job Factory Service (MMJFS). MMJFS validates global user identity mapped to a workspace.
6. The Grid middleware contacts the appropriate job manager (Condor, Fork, PBS etc.) that allocates resources and executes the parallel or sequential legacy code on the Compute Servers.
7. As long as the client credentials have not expired and the *GLCProcess* is still alive, the client can contact GEMLCA for checking job status and retrieving partial or final results at any time.

3.2 Resource & Service Monitoring

In order to offer GEMLCA legacy code services for production Grid systems, automatic testing of these services is inevitable. The GEMLCA Monitoring Toolkit (GMT) was developed to provide monitoring information based on probes (scripts) checking the status of GEMLCA resources. Using the GMT, system administrators are automatically alarmed when a test fails and can also request the execution of any test on-demand. The GMT also assists P-GRADE portal users when mapping the execution of workflow components to resources by offering only verified Grid resources when creating a new workflow or when rescuing a failed one.

The GMT is based on MDS4 (Monitoring and Discovery System) that is part of the Globus distribution. MDS4 is capable to collect, store and index information about resources, respond to queries concerning the stored information using the XPath language, and control the execution of testing and information retrieval tools built as part of the GEMLCA Monitoring Toolkit. It can be extended and tailored to obtain specific information by means of polling resources, subscription to obtain notifications regarding changes to the state of specific resources, and execution of test and information collection probes.

As part of the GMT, several probes were implemented that collect information concerning the state of basic Globus services, local job manager functionality, and GEMLCA services. The probes can immediately be used as standalone tools executed automatically from the MDS by means of an XML configuration file, or manually from a command line interface, and they are also integrated into the P-GRADE portal assisting both system administrators and end-users.

System administrators can configure the MDS4 service to run the various probes at pre-defined intervals. The results are collected by a portlet that is integrated into the P-GRADE portal. Administrators can also select a specific probe from a drop-down list displayed by a portlet and run it to verify the state of a specific service at a specific site on demand. GMT probes can also assist end-users when mapping a new workflow execution onto available Grid resources, or when rescuing and re-mapping a failed workflow. In the latest P-GRADE portal release mapping of workflow components to underlying resources is done either manually by the end-user, or by a broker, for example in EGEE by the LCG broker. The GMT aims to support manual mapping, when no broker is available, by dynamically querying the MDS4 during workflow creation time, and

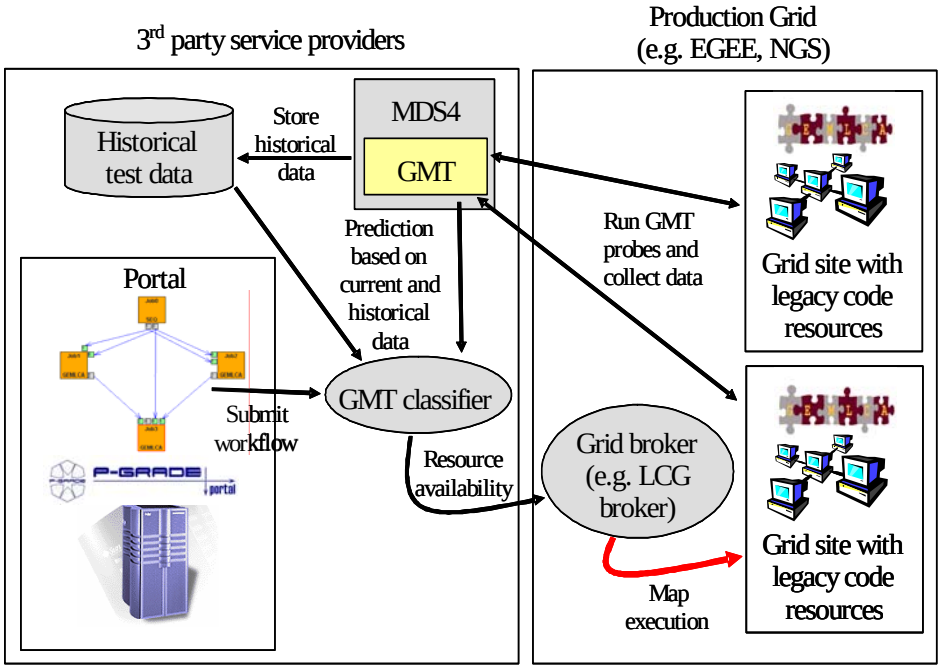


Figure 3: GMT based resource availability prediction

offering only those GEMLCA resources for mapping those latest GMT test results were positive. Although, this does not guarantee that the resource will actually work when executing the workflow, the probability of a successful execution will be significantly increased. Work is also undergoing to connect the GMT to the LCG resource broker, as illustrated on Fig. 3. GMT, as shown on the figure, runs probes on the production Grid resources and, besides updating the MDS indexing service, also creates a historical database. When the portal submits a workflow, a classifier component runs data mining algorithms on this historical data and determines which resources are "very likely to be alive". This information can be passed to the production Grid broker, for example in case of an LCG broker within the JDL (Job Description Language) file. The broker then maps the execution to the appropriate resources taking now the GMT provided information into consideration too.

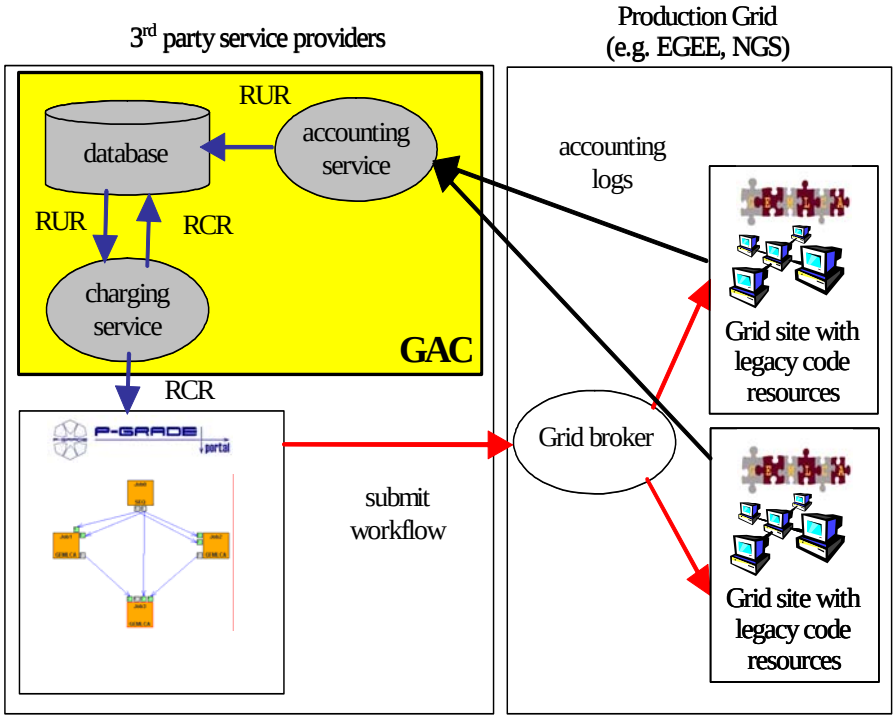


Figure 4: Grid resource usage accounting and charging

3.3 GEMLCA Accounting And Charging

After analysing existing accounting and charging solutions, such as DGAS [20], GREAC [21], GridBank [22] and taking into consideration accounting, brokering and charging requirements, business models and usage scenarios, a service-oriented architecture, the Grid Accounting and Charging architecture (GAC) has been elaborated for accounting and charging for usage of GEMLCA resources and services. The architecture, given on Fig. 4, incorporates the accounting service, the charging service, the GAC database with accounts and usage records, and portlets to manage accounting and charging. There are two players, who use GAC; consumers, i.e. Grid users, and providers, i.e. Grid resource and service providers. Both players have accounts in the GAC database. GAC is built on the Resource Usage Record (RUR) [23]. RUR, which is an XML document, contains the components given in Table 1.

GEMLCA generates accounting logs that contain the job data, plus the resource and user IDs (or certificates). The resource and user data is stored in the GAC database in order to minimise the size of the accounting log. The GAC accounting service has two main tasks: first, it manages accounts, i.e. it opens, closes accounts, retrieves, displays and updates accounts' data, and handles transactions, i.e. it deposits and withdraws money to/from accounts. Secondly, it processes the accounting log of each job submission and service request, defines the usage time, and creates

and stores the RUR in the corresponding resource and user accounts.

resource data	user data
host name / IP address	host name / IP address
certificate name	certificate name
host type	
local job ID	
wall clock time + price	
user CPU time + price	**job data**
system CPU time + price	job ID
main memory + price	application name
secondary storage + price	job start date
I/O channels + price	job end date

Table 1: Resource Usage Record

The charging service uses the RUR as input data to generate the Resource Charge Record (RCR), which is given in Table 2. The charging service calculates charges according to the business model of the resource and service. The business model defines how to charge for usage, how to pay, etc. The charging service should be able to support different charging policies, such as "pay before use", "pay after use", "pay as you go", etc.

user certificate name
resource certificate name
job ID
resource business model
resource usage date
resource usage time
amount to be charged

Table 2: Resource Charge Record

In order to charge for GEMLCA service usage, GEMLCA legacy code services have to be extended with economic service data. This economic service data includes attributes like pricing mechanism and currency, and additional information like liability, compensation and refund policies. As GEMLCA describes the legacy codes in the XML-based Legacy Code Interface Description file (LCID), this file can be extended with the economic data.

The charging service stores the RCR in the GAC database and also sends the RCR to both the resource/service provider's and the user's accounts. Having the RCR the charging service transfers the calculated amount from the user's account to the provider's account.

To provide access to the providers' and users' accounts and manage them, an account portlet will be added to the GEMLCA portal.

4. Conclusions and Further Work

Current production Grid systems serve only academic communities and do not charge for their services. However, extending the usage of these Grids towards businesses and industry is the next inevitable step for the widespread take-up of Grid computing. This step requires several enhancements of current Grids. Our paper described some necessary extensions to the current GEMLCA architecture in order to support these next generation Grids. User support services, like GEMLCA, are inevitable for future Grid systems and will enable the real usability of these infrastructures. Providing these pioneering developments in GEMLCA we are intending to create reference implementation for similar user support services.

This paper concentrated on the latest ongoing research and development work that extends the architecture with dynamic user management, resource monitoring and availability prediction, and accounting and charging capabilities. Most of these concepts are already implemented as prototypes and their full integration into the GEMLCA architecture is currently work in progress.

References

[1] T. Delaittre, T. Kiss, A. Goyeneche, G. Terstyanszky, S.Winter, P. Kacsuk: GEMLCA: Running Legacy Code Applications as Grid Services, Journal of Grid Computing Vol. 3. No. 1-2. June 2005, Springer Science + Business Media B.V., Formerly Kluwer Academic Publishers B.V. ISSN: 1570-7873, pp 75-90

[2] A.Goyeneche, T.Kiss, G.Terstyanszky, G.Kecskemeti, T.Delaitre, P.Kacsuk, S.C. Winter, Experiences with Deploying Legacy Code Applications as Grid Services using GEMLCA, Conf. Proc. of the European Grid Conference, February 14 -16, 2005, Science Park Amsterdam,The Netherlands, Volume editors: P.M.A. Sloot, A.G. Hoekstra, T. Priol, A. Reinefeld, M. Bubak, pp 851-860, ISBN: 3-540-26918-5

[3] A. Tarczynski, T.Kiss, D. Qu, G. Terstyanszky, T. Delaittre, S. Winter, Application of Grid Computing for Designing a Class of Optimal

Periodic Nonuniform Sampling Sequences, Conf. Proc. of the Grid-Enabling Legacy Applications and Supporting End Users Workshop, within the framework of the 15th IEEE International Symposium on High Performance Distributed Computing , HPDC'15, Paris, France, June 19-23, 2006

[4] The UK National Grid Service Website,http://www.ngs.ac.uk/

[5] The WestFocus Grid Alliance Website, http://www.gridalliance.co.uk/

[6] The TeraGrid Website, http://www.teragrid.org

[7] The Open Science Grid Website, http://www.opensciencegrid.org/

[8] The EGEE Website, http://public.eu-egee.org/

[9] The Globus Toolkit GT2, http://www.globus.org/

[10] EGEE gLite version 1.5 Documentation, http://glite.web.cern.ch/glite/documentation/default.asp

[11] The Globus Toolkit GT4, http://www.globus.org/

[12] Y. Huang, I. Taylor, D. W. Walker, "Wrapping Legacy Codes for Grid-Based Applications", Proceedings of the 17th International Parallel and Distributed Processing Symposium, workshop on Java for HPC), 22-26 April 2003, Nice, France. ISBN 0-7695-1926-1

[13] B. Balis, M. Bubak, M. Wegiel. A Solution for Adapting Legacy Code as Web Services. In Proceedings of Workshop on Component Models and Systems for Grid Applications. 18th Annual ACM International Conference on Supercomputing, Saint-Malo, July 2004

[14] D. Gannon, S. Krishnan, A. Slominski, G. Kandaswamy, L. Fang, "Building Applications from a Web Service based Component Architecture, in "Component Models and Systems for Grid Applications" edited by V. Getov and T. Kiellmann, Springer, 2005, pp 3-17, ISBN 0-387-23351-2.

[15] P. Kacsuk G. Sipos: Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal, Journal of Grid Computing, , Springer Science + Business Media B.V., Vol. 3 No. 3-4 Dec, 2005, pp 221-238, ISSN: 1570-7873

[16] J. Novotny, M. Russell, O. Wehrens: GridSphere, "An Advanced Portal Framework", Conf. Proc. of the 30th EUROMICRO Conference, August 31st - September 3rd 2004, Rennes, France.

[17] James Frey, Condor DAGMan: Handling Inter-Job Dependencies, http://www.bo.infn.it/calcolo/condor/dagman/

[18] Globus Workspace Management Service, http://www.globus.org/

[19] G.Terstyansky, T. Delaitre, A. Goyeneche, T, Kiss, K. Sajadah,S.C.Winter, P.Kacsuk, Security Mechanisms for Legacy Code Applications in GT3 Environment, Conf. Proc. of the 13th Euromicro Conference on Parallel, Distributed and Network-based Processing, Lugano, Switzerland, February 9-11, 2005

[20] Andrea Guarise: Grid Accounting in EGEE using DGAS. Current practices Terena Networking Conference, Poznan, June 8, 2005

[21] Shawn Mullen, IBM, Matt Crawford, FNAL, Markus Lorch, VT, Dane Skow, FNAL – "Site Requirements for Grid Authentication, Authorization and Accounting" GFD-I.032 – GGF, October 13, 2004

[22] Alexander Barmouta, Rajkumar Buyya: GridBank: A Grid Accounting Services Architecture (GASA) for Distributed System Sharing and Integration

[23] RUR, Resource Usage Record Working Group, Global Grid Forum, http://www.gridforum.org/3_SRM/ur.htm

Meeting the Design Challenges of Nano-CMOS Electronics: An Introduction to an Upcoming EPSRC Pilot Project

R. Sinnott[1], A. Asenov[2], D. Berry[3], D. Cumming[2], S. Furber[4], C. Millar[2], A. Murray[5], S. Pickles[6], S. Roy[2], A. Tyrrell[7], M. Zwolinksi[8]

[1]National e-Science Centre, University of Glasgow
[2]Department of Electronics and Electrical Engineering, University of Glasgow
[3]National e-Science Centre, University of Edinburgh
[4]Advanced Processor Technologies Group, University of Manchester
[5] Mixed Mode Design Group, University of Edinburgh
[6]e-Science North West, University of Manchester
[7]Intelligent Systems Group, University of York
[8]Electronic Systems Design Group, University of Southampton
ros@dcs.gla.ac.uk

Abstract

The years of 'happy scaling' are over and the fundamental challenges that the semiconductor industry faces, at both technology and device level, will impinge deeply upon the design of future integrated circuits and systems. This paper provides an introduction to these challenges and gives an overview of the Grid infrastructure that will be developed as part of a recently funded EPSRC pilot project to address them, and we hope, which will revolutionise the electronics design industry.

1. Introduction

Progressive scaling of complementary metal oxide semiconductor (CMOS) transistors, as tracked by the International Technology Roadmap for Semiconductors (ITRS) [1] and captured in Moore's law, has driven the phenomenal success of the semiconductor industry, delivering larger, faster, cheaper circuits. Silicon technology has now entered the nano-CMOS era with 40 nm MOSFETs in mass production at the current 90 nm ITRS technology node [2] and sub-10 nm transistors expected at the 22 nm technology node, scheduled for production in 2018. 4 nm transistors have already been demonstrated experimentally [3], highlighting silicon's potential for scaling beyond the end of the current ITRS. However, it is widely recognised that variability in device characteristics and the need to introduce novel device architectures represent major challenges to scaling and integration for present and next generation nano-CMOS transistors and circuits. This will in turn demand revolutionary changes in the way in which future integrated circuits and systems are designed. To tackle this problem, strong links must be established between circuit design, system design and fundamental device technology to allow circuits and systems to accommodate the individual behaviour of every transistor on a chip. Design paradigms must change to accommodate this increasing variability. Adjusting for new device architectures and device variability will add significant complexity to the design process, requiring orchestration of a broad spectrum of design tools by geographically distributed teams of device experts, circuit and system designers. This can only be achieved by embedding e-Science technology and know-how across the whole nano-CMOS electronics design process and revolutionising the way in which these disparate groups currently work. The recently funded "*Meeting the Design Challenges of Nano-CMOS Electronics*" EPSRC pilot project is looking directly at building a Grid infrastructure that will meet the challenges raised by the scaling problems across the whole of the electronics industry. This 4-year project is expected to start in October 2006 hence this paper is focused upon the domain requirements and scientific challenges that will shape the Grid infrastructure. We also present initial ideas in the design and implementation of the Grid infrastructure that will address the specific challenges of this domain.

The rest of the paper is structured as follows. Section 2 focuses on the scientific demands of the nano-CMOS electronics area and the problems arising from decreasing transistor scalability. Section 3 gives an overview of the demands that this domain places on the Grid infrastructure to be developed. Section 4 focuses on initial ideas on the design and development of this infrastructure, and we conclude with a summary of our plans for the future.

2. Scientific Challenges

The rapid increase in intrinsic parameter fluctuations represents the most serious challenge facing the electronics industry today. These fluctuations stem from the fundamental discreteness of charge and matter. They are fundamental, truly stochastic and cannot be eliminated by tighter process control. The major sources of intrinsic parameter fluctuations include random discrete dopants (Fig. 1 and Fig. 2), line edge roughness and oxide thickness fluctuations [4].

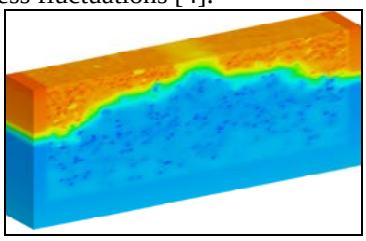


Fig. 1 Random discrete dopants in a 35 nm MOSFET from the present 90 nm technology node

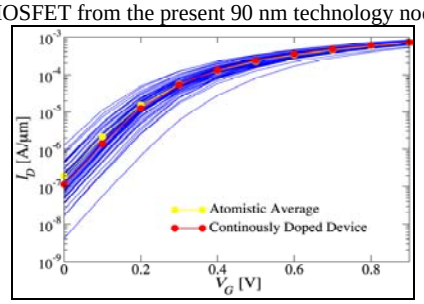


Fig. 2 Corresponding variations in the current-voltage characteristics of 200 transistors with different dopant distributions

While intrinsic parameter fluctuations and resultant device mismatch have hitherto affected only analogue design, they now challenge the power consumption, yield and reliability of digital circuits [5]. One of the first digital "casualties" is SRAM, which occupies significant real estate in current System On Chip (SoC) devices [6]. Fig. 3 illustrates the random dopant induced distribution of static noise margin in an ensemble of SRAM cells of various cell ratios at the transition between the 90 nm and 65 nm technology nodes.

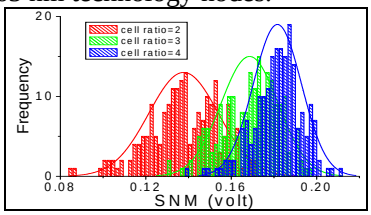


Fig. 3 Corresponding distribution of the static noise margins in 6T SRAM cells

Only a large cell ratio can produce acceptable yield in the presence of fluctuations, increasing cell area and reducing the benefits of scaling. Thus, variability already causes significant circuit and system challenges at a time when design margins are shrinking, owing to lowered VDD and increased transistor count. Exponentially increasing design difficulties require novel statistical design solutions. This is exacerbated by the enormous logistic, computational and data management needs of statistical design techniques which thus represents a prime candidate for exploitation of Grid technology.

In the years of 'happy scaling', circuit and system design used conventional bulk MOSFETs that behaved remarkably similarly over many technology node generations. Variability was associated with fabrication processes and equipment-related non-uniformities. Differences in, for example, implantation dosage and lithographical alignment were responsible for wafer-to-wafer parameter variations, and on-wafer non-uniformities were responsible for on-wafer variations. Simple workstation based 'corner' analysis was able to assess the impact of variations in the design process. As a result, compact models extracted from measured device characteristics supported by simple rule-based tools allowed a high level of abstraction, distancing circuit and systems designers from device design and technology.

New types of device parameter variations, related to the introduction of sub-wavelength lithography and process induced strain, emerged in the transition from the 130 to 90 nm technology nodes. These now play an increasingly important role. Optical proximity correction (OPC) and phase shift masks result in variations in the shape of transistors with otherwise identically drawn gate layouts, depending upon the surrounding cell topology. These dimensional variations are commensurable with the gate length and can result in significant changes in transistor characteristics. Compressive and tensile strain, induced typically by SiGe source/drain regions and Si_3N_4 blankets respectively, were introduced at the 90 nm

node to improve p- and n-MOSFET mobility and performance [2]. The strain distribution and device performance are determined by not only gate topology, but also by the gate-to-gate spacing, the width of the source/drain regions, the position and shape of the contact windows and the distance to the shallow trench isolation. OPC and strain-induced variations at and beyond the 65 nm node mean that standard design rules and conventional physical verification may not be sufficient to ensure yield without an unacceptable degradation in cell density. At the 45 nm technology node, hundreds of pages of design rules are expected to replace the traditional single page of rules, in order to maintain yield. Variations must be considered early in the design flow. Furthermore, the strong link between circuit and device design and underpinning technology design that was broken, for good reasons, in the early days of VLSI design must be re-established.

It is expected that there will be no single replacement for conventional MOSFETs and that disparate device architectures will coexist and compete. All new device architectures require a more-or-less new design approach, altering device and circuit layout and the electrical behaviour of each generation of nano-CMOS devices. This adds to the design challenges associated with increasing device and circuit variability.

Grid technologies when correctly applied across the nano-CMOS electronics design space can address these challenges. These infrastructures should allow designers to choose the most appropriate technologies for each design project, with the resources needed to deliver optimal, competitive design solutions. Importantly in this domain (which is one of the distinguishing features from other domains) is the importance of intellectual property (IP). IP for designs, data and processes is fundamental to this domain and SMEs and collaborators must be assured that the security infrastructure supporting the new design processes fully protects IP integrity.

3. Grid Challenges

Whilst there are numerous areas where we expect to extend state of the art in Grid know-how, our fundamental goal is to facilitate scientific research: ideally to revolutionise the electronics design industry.

We have identified several areas that characterise capabilities of the infrastructure needed to support the scientific challenges of the nano-CMOS domain. For each of these areas we envisage developing a family of components comprising frameworks that can be applied by the scientists for their daily research, incorporating all aspects of the designs of circuits and systems incorporating the decreasing scaling and expanding design capabilities that face the electronics industry described previously.

Specifically in exploratory domain discussions on capabilities needed by the electronics design protagonists we have identified the following four key areas that are crucial to the success of the Grid infrastructure: workflows, distributed security, data management, and resource management.

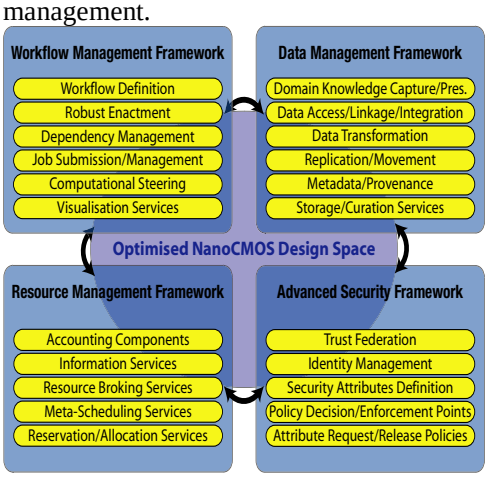


Figure 4: Framework and Components of the Grid Infrastructure

We consider these in turn and why they are important, and provide an initial overview of our intentions in delivering these components.

3.1 Workflow Management Framework

The definition and enactment of workflows suitable for the nano-CMOS electronics domain will form the cornerstone of our work. This will require the wrapping of existing simulation software as Grid services by the application design groups, aided by the e-

Science partners. Figure 5 indicates our initial expectations on the kinds of workflow components that need to be supported.

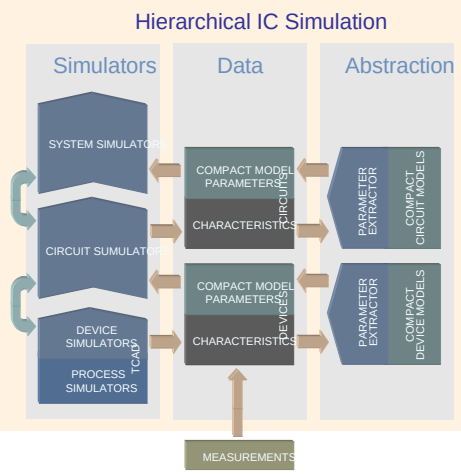


Fig. 5 Hierarchical simulation methodology needed to capture the impact of variability on design.

We recognise already an important issue here will be to describe the services and data sets with sufficient information (meta-data) to allow for their subsequent discovery, access and usage in the design workflows. Given the multitude of atomistic device simulation software variations, where each simulation is compiled to explore slightly different parameter sets, ensuring that the similarities and distinctions between these simulations and their input/output data sets are captured is fundamental to the workflow definition and their enactment. Where possible, the design groups will automatically capture this meta-data, building on existing tools and expertise from the e-Science partners as described below in section 3.3.

We plan to allow design groups to work together and develop libraries of workflows that allow other scientists to run, and subsequently manage, multiple, concurrently executing simulations. The scientists will be able to browse and search for workflows, designs and data sets relevant to their work, subject to their security attributes.

We plan to allow for broad parameter sweep simulations which, through user interaction, can be refined to fine-grained parameter explorations when a situation of specific scientific interest arises. It will also allow for the efficient overall usage of the Grid infrastructure resources, since uninteresting simulations can be stopped or steered to more interesting regions of parameter space. Services supporting visualisation of scientific data will be integrated into these workflows. Interactive steering in a workflow context will be a novel application of RealityGrid (www.realitygrid.org.uk) and myGrid (www.mygrid.ork.uk) software, and their integration will give rise to new requirements on both. As the RealityGrid software is maintained by project partners, we can deal with requirements on it ourselves. However, future development of myGrid software is the responsibility of OMII-UK (www.omii.ac.uk), and requirements on the myGrid provenance service and Taverna workflow enactment engine or their successors must be considered in a broader context. In order to manage this external dependency, we will establish an ongoing dialog with OMII-UK at the project start, negotiate mechanisms for feeding our requirements into the OMII-UK roadmap, and if necessary, contribute effort to their realisation and verification.

3.2 Resource Management Framework

The optimised use of the compute resources requires up to date information on the status of the Grid infrastructure. To support this, the e-Science partners will deploy Grid services which will capture near real time data on the status of the Grid infrastructure and the associated Grid services deployed. We will adapt and extend existing services, e.g. from the NeSC BRIDGES project (www.nesc.ac.uk/hub/projects/bridges) and the Globus Alliance web service based monitoring and discovery system (MDS) [7], while taking into account developments on the National Grid Service (NGS www.ngs.ac.uk). These services will include capabilities for publishing and subscribing to information service data sets, for filtering of the associated data with these services and for storing and archiving the data associated with these services. Aggregation of such

information will be supported and incorporated within the workflow management framework to influence real time workflow enactment. The definition of appropriate schemas will be fundamental to this work and we will build on the OGSA working group's current comparison of GLUE and DMTF CIM, where the e-Science partners are intimately involved [8].

Understanding where a given simulation should be executed can be an involved process involving numerous metrics, e.g. the status of the Grid infrastructure at that time, the chip architecture that the code has been compiled for, the location of specific data sets, the expected duration of the job, the authorisation credentials of the end user wishing to run the workflow etc. To support the scientific needs of the project, we will survey on-going work in this area, and if necessary develop and deploy our own meta-scheduling and planning services building on the basic meta-scheduler currently supported in the NeSC BRIDGES project.

Metrics on data movement, as well as existing and predicted resource usage will be explored as part of this work. This will include the exploration of different economic models, such as maximal job throughput, minimal job cost in the presence of separately owned and managed Grid resources, each with their own specific accounting and charging policies.

Given that the fEC model of research funding requires monies to be set aside for time spent on major HPC clusters, we propose to explore existing state of the art in Grid based accounting services. One of these which we will explore and potentially enhance is the Resource Usage Service [9] which is currently being considered for deployment across the NGS, and will be considered for the (www.scotgrid.ac.uk) and the Northwest Grid. We plan here to draw on expertise and software from the Market for Computational Services project (see http://www.lesc.ic.ac.uk/markets/). In the latter part of the project, we also plan to explore advanced resource broking, reservation and allocation. We would expect to learn from and feed into the GRAAP-WG at GGF and on-going efforts in this area, such as the WS-Agreement standard specification [10]. The case for advanced reservation and allocation will be tempered by the practical impact on reduction of overall utilisation of the compute resources and the associated cost impact this will incur. A better understanding of these issues is crucial in the fEC era.

3.3 Data Management Framework

The data sets that are generated by device modellers and circuit/systems designers are significant, both in size and in number as indicated in table 1.

Task	Accumulated data project lifetime
SPICE cell & cct char. T-level sim. (nanosim) Gate-level sim. (Verilog) Behavioural sim. (Sys C) Extraction	10GB 5-10TB 1TB 100GB 20GB
3D TCAD sim. 3D 'atomistic' sim. Compact models	1TB 5-10TB 30GB
Circuit level fault sim. Behavioural sim.	100GB 10GB
Evolutionary systems	30GB
Fragment sims. Cells sims. Extraction to STA	10 TB 5 TB 100 GB
Sim. mixed-mode circuits with noise and variation.	5TB

Table 1: Summary of Expected Data Set Accumulation

The tight linkage and integration of these data sets and models is paramount. We plan to use and extend the OGSA-DAI system (www.ogsadai.org.uk) and the OGSA Data Architecture efforts to meet this need, including the current work in OGSA-DAI to manage files as well as databases. Particular areas where we will focus will include: (i) the integration of OGSA-DAI with workflow systems including data transformation capabilities, (ii) the development of appropriate meta-data schemas specific to the electronics domain, (iii) attaching security restrictions to data as it is moved (rather than to the sources and sinks themselves), (iv) comparison of remote data access with

pre-staging or the movement of application code to the data, (v) the efficient specification of data transfer for a variety of endpoints (e.g. files, query results, in-memory data sets), (vi) the integration of OGSA-DAI and provenance systems.

Annotating data from simulations with appropriate meta-data will allow for future tracking and longer-term curation and will form a fundamental part of the data management framework. Building on the close synergy of NeSC with the National Digital Curation Centre (DCC www.dcc.ac.uk) we will exploit direct expertise in how best to capture and subsequently exploit such information.

Some of the data sets, software and processes will be of commercially sensitive nature, and access to them must be restricted to suitably authorized individuals; when such data is transported or replicated, it must be done so securely, and in extreme cases, the data may not be replicated outside the domain in which it originated. Our data management framework will be closely coupled with the security framework to ensure IP is handled appropriately.

3.4 Advanced Security Framework

Novel device designs and their potential impact on integrated systems give rise to highly sensitive, commercial exploitation possibilities. Without a robust, reliable and simple Grid security infrastructure (from the end user perspective) incorporating very fine grained security capabilities, the electronics design community, from SMEs to major corporations involved in the electronics industry, will not involve themselves. The widespread acceptance and uptake of Grid technology can only be achieved, if it can be demonstrated that security mechanisms needed to support Grid based collaborations are at least as strong as local security mechanisms.

Drawing on the e-Health projects at NeSC, we will show how data security, confidentiality and rights management can be supported by the Grid infrastructure to protect commercial IP interests.

The predominant, current, method by which security is addressed in the Grid community is through Public Key Infrastructures (PKI) to support authentication. This addresses user identity issues, but for the fine-grained control required over users' privileges on remote resources, we require advanced authorisation services. The project partners bring a wealth of experience in the practical establishment and management of advanced privilege management infrastructures using the latest advances in solutions such as PERMIS (www.permis.org). Additionally, the UK academic community is in the process of deploying the Internet2 Shibboleth technologies (shibboleth.internet2.edu) to support local (existing) methods of authentication for remote login to resources. Within this proposal we will explore the use of Shibboleth technologies to simplify the overall end user experience of access to, and usage of, Grid resources, drawing upon numerous other projects. We will identify and define the security attributes required in the electronics design domain. A direct application of this security infrastructure will be to restrict access to, and usage of, data sets and software which have IP restrictions. These are novel challenges and remain open issues to be solved within the Grid community.

3.5 Grid Summary

The seamless orchestration of the four frameworks and their components will create *a virtual nano-CMOS design foundry* where the behaviour of advanced systems and circuits can be predicted based upon, and feeding back into, device models and processes. The Grid infrastructure will allow the exploration of interesting challenges arising from situations where designs have been identified as invalid, erroneous or superseded. Tracking data sets and relations between the design space and models, whilst keeping the data design space as accurate as possible, is novel research in itself. Further challenges will be encountered in ensuring that IP issues and associated policies are demonstrably enforced by the infrastructure. The results of this work will directly impact upon future Grid efforts in the standardisation

and implementation areas. We expect to directly input the security solutions incorporating Shibboleth and advanced authorisation into OMII-UK version 5 releases (currently scheduled for 2007 in the draft roadmap) and provide a rigorous assessment and feedback on their workflow and enactment engine and their enhancements.

4. Initial Design Scenarios

To understand how these frameworks and components will be applied to support the nanoCMOS Grid infrastructure we outline one of the key scenarios that we intend to support. The requirements capture, design and prototyping phases that run through the lifetime of the project will refine this scenario and produce numerous other scenarios.

We consider a scenario where a scientist wishes to run an atomistic device modelling simulation based on the commercial Taurus software which requires a software licence to generate statistically significant sets of I/V curves. These I/V curves will then be fed into the Aurora software to generate compact models which will subsequently be fed into a circuit analysis software package such as SPICE. At each of these stages the scientists will be able to see the results of the software production runs and influence their behaviour. Note that Taurus, Aurora and SPICE are representative examples only and a far richer set of simulation software will be supported.

Step 1: A Glasgow researcher attempts to log in to the project portal and is automatically redirected via Shibboleth to a WAYF service (we will utilise the SDSS Federation (www.sdss.ac.uk)) and authenticates themselves at their home institution with their home username/password (denoted here by LDAP server – as used at Glasgow).

Step 2: The security attributes agreed within the nanoCMOS federation are returned and used to configure the portal, i.e. they see the services (portlets which will access the Grid services) they are authorised for. These security attributes will include licenses for software they possess at their home institution and their role in the federation amongst others.

Step3: A client portlet for the Taurus software is selected by the scientist.

Step 4: The scientist then explores and accesses the virtualised data design space for input data sets to the Taurus software production run. This might consist of experimental data, first principle simulation data or data stemming from circuit or system inputs. Once again, what the scientist can see is determined by their security privileges (to protect IP). The meta-data describing the characteristics of the data such as the confidence rating (whether it has been validated or been superseded), who created it, when it was created, what software (or software pipelines) and which versions of the software were used to create the data, whether there are any IP issues associated with the data will all be crucial here.

Step 5: Once the user has selected the appropriate data sets needed for generation of the appropriate I/V curves, the meta-scheduler/planner is contacted to submit the job. Where the jobs are submitted will depends on which sites have access to an appropriate license for the Taurus software as well as the existing information on the state of the Grid at that time.

Step 6: Once the meta-scheduler/planner submits the jobs to the Grid resources and the portlet is updated with real time information on the status of the jobs that have been submitted (whether they are queued, running, completed). The actual job submission might be involved here, for example when the input files are very large and require to be partitioned. We will draw on the JSDL standards work here, e.g. through the OMII GridSAM software.

Step 7: On completion (or during the running of the Taurus simulations), the resultant I/V data sets are either stored along with all appropriate meta-data (not shown here) or fed directly (potentially via appropriate data format transformation services not shown here) into the Aurora software which in turn will decide where best to run the Aurora simulation jobs using the meta-scheduler/planner and available run time information.

Step 8: The Aurora client portlet will allow for monitoring of the resultant compact models and allow these to be fed

into the SPICE models (also started and run using the meta-scheduler/planner).

This orchestration of the different simulations and how they can feed directly into one another typifies the capabilities of the Grid infrastructure we will support and indicates how we will support a virtual nanoCMOS electronics design space. The e-Science partners in the project will initially design families of such workflows which the scientists can parameterise and use for their daily research, however as the scientists become more experienced in the Grid technologies, they will design and make available their own workflows for others to use.

Interesting challenges that arise from this domain that the Grid infrastructure will allow to explore will be when designs have been recognised to be invalid, erroneous or superseded. Tracking data sets and relations between the design space and models to keep the data design space as accurate as possible is novel. Further key challenges will be to ensure that IP issues and associated policies are demonstrably enforced by the infrastructure. Drawing on the e-Health projects at NeSC, we will show how data security, confidentiality and rights management can be supported by the Grid infrastructure to protect commercial IP interests.

5. Conclusions

The electronics design industry is facing numerous challenges caused by the decreasing scale of transistors and the increase in device design flexibility. The challenges whilst great are not insurmountable. We believe that through a Grid infrastructure and associated know-how, a radical change in the design practices of the electronic design teams can be achieved.

To address these challenges, the scientists cannot work in isolation, but must address the issues in circuit and systems design in conjunction with the atomic level aspects of nano-scale transistors. We also emphasise that the success of this project will not be based upon development of a Grid infrastructure alone. It requires the electronics design community "as a whole" to engage. This can only be achieved if they are integrally involved in the design of this infrastructure. To achieve this we plan to educate the electronics design teams to an extent whereby they can Grid enable their own software, design their own workflows, annotate their own data sets etc. It is only by the successful adoption of these kinds of practices that the infrastructure will "revolutionise the electronics design industry" as we hope.

6. References

1. International Technology Roadmap for Semiconductors Sematech *http://public.itrs.net*

2. R. Khumakear at al., "An enhanced 90nm High Performance technology with Strong Performance Improvement from Stress and Mobility Increase through Simple Process Changes" 2004 Symposium on VLSI Technology, Digest of Technical Papers, pp 162-163, 2004

3. H. Wakabayashi, "Sub 10-nm Planar-Bulk-CMOS Device using Lateral Junction Control", IEDM Tech. Digest, pp. 989-991, 2003.

4. A. Asenov, A. R. Brown, J. H. Davies, S. Kaya and G. Slavcheva, "Simulation of Intrinsic Parameter Fluctuations in Decananometre and Nanometre scale MOSFETs", IEEE Trans. on Electron Devices, Vol.50, No.9, pp.1837-1852, 2003.

5. P.A. Stolk, H.P. Tuinhout, R. Duffy, et al., "CMOS Device Optimisation for Mixed-Signal Technologies", IEDM Tech Digest, pp.215-218, 2001

6. B. Cheng, S. Roy, G. Roy, F. Adamu-Lema and A. Asenov, "Impact of Intrinsic Parameter Fluctuations in Decanano MOSFETs on Yield and Functionality of SRAM Cells", Solid-State Electronics, Vol. 49, pp.740-746, 2005.

7. Globus Toolkit Monitoring and Discovery System, www.globus.org/toolkit/mds

8. Open Grid Service Architecture Common Information Model, www.ggf.org/cim

9. Resource Usage Service Working Group, www.doc.ic.ac.uk/~sjn5/GGF/rus-wg.html

10. A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. Web services agreement specification WS-Agreement (draft), 2004.

Types of grid users and the Customer-Service Provider relationship: a future picture of grid use

A paper for the UK e-Science All Hands Meeting, September 2006

Mark Norman,
University of Oxford

1. Abstract

Who will be the grid users of tomorrow? We propose a categorisation of 'future grid' users into the following categories: Service End-User, Power User (with three distinct sub-types), Service Provider and Infrastructure Sysadmin. A further basic type could be argued as Third Party Beneficiary. This paper outlines the possible characteristics of these 'types' of users. For users that have layers of applications or, for example, a portal between them and the grid resource, it is almost certain that heavyweight security solutions, as we have with client digital certificates, are too onerous and unnecessary. It is likely that some users will, however, need client digital certificates, due to the level of control that they may exert on individual grid resources. We also outline a Customer-Service Provider model of grid use. It may be that authentication and authorisation for the SEU 'customers' should be the responsibility of the Service Providers (SPs). This would hint at a more legal framework for delegating authority to enable grid use, but one which could be more secure and easier to administer. Such a model could also simplify the challenges of accounting on grids, leaving much of this onerous task to the Service Providers.

2. Introduction

2.1. The ESP-GRID project

The Evaluation of Shibboleth and PKI for Grids (ESP-GRID) project's central aim was to achieve a deeper understanding of the potential role that Shibboleth can play in grid authentication, authorisation and security. One of the main outcomes of the project has been that Shibboleth is applicable to some users in some situation and client-based PKI is applicable largely to more technical users in other situations. This gave rise to an examination of the future types of grid users. This arose from a series of brainstorming and consultation sessions with current grid users and developers. The inking within this paper represents some of this output supported by anecdotal evidence and findings in the literature.

2.2. When will the grid be really useful?

Before Netscape's browser, Mosaic, was given away free in 1994, the Internet was the domain of the educated and technically knowledgeable. Even within that educated elite, the use of the Internet was dominated by a few research subject areas, possibly arenas in which the development of computing itself had been highly relevant for many years. What changed? The introduction of a graphical interface that was easier to use and was more intuitive did increase the rate of uptake of home computing.

Many interested groups must hope that grid technology must be approaching the metaphoric 'release of the browser' stage some time soon. Whether there will be a surge in take-up, as seen with Internet technologies after 1994, or whether it will be a more steady increase remains to be seen. However, it is the availability and ease of use to the greater community that will make the breakthrough. This paper is focussed mainly upon the educational and research use of grid technology. The engagement of the average citizen with grid technology will take much longer. We believe that the experience of take-up

of the Internet is relevant to the divide between 'researchers experienced in programming or scripting' and the 'rest' of the research community.

2.3. Are we talking to the right users?

Anecdotal evidence of researchers refusing to engage and benefit from grid technology suggests that when an application interface is presented that is easy to use, the uptake is strong (e.g. Sinnott, 2006). As the Market for Computational Services Project notes, the inability to use a simple 'service' such as a resource broker in itself leads to a lack of ease of use and little motivation for the end user leading to little or no take-up for real use (Grid Markets, 2003).

Authors have previously noted that the current grid middleware is too intimidating for many users, and have often focussed on the security aspects (e.g. Beckles, 2004a). These aspects are important as they are often the most onerous for the non-computer specialist. In temporary lieu of the work, noted above, to collect requirements from current non-users (Beckles, 2004b), we believe that we should examine the types of users that are emerging within grid computing and consider their generic security profiles as well as their likely access management requirements. This may assist in identifying such users in order to carry out a real world requirements analysis. However, until such an analysis is made, our work is merely a guide to the likely categories of users.

We have, by necessity, very technical users at present. This may distract us from building an accessible grid for future users who may be far less computing-technical.

The following sections of this paper present our view of these users of tomorrow. This is a personal view, based partly on recent experience within the ESP-GRID project and partly on predictions arising from the use of the Internet and the Web.

3. Types of grid users

3.1. Categories of users

Table 1 presents a summary of the types of grid users that exist, or that will exist in the very near future. Clearly, as with any

Table 1 Grid users of the future

Type of user	Typical characteristic	Main role
SEUD	Service End-User (data). Little or no computing expertise.	User of applications served by SPs. Uploads data or runs queries.
SEUX	Service End User (executables). Some understanding of code creation.	As SEUD, but runs either executable code or scripts via SPs
PUA	Power User Agnostic of grid resource node. High degree of computing expertise.	Develops programs and data but does not care where processing takes place.
PUS	Power User requiring Specific grid resource nodes. High degree of computing expertise.	As PUA but may have more platform etc. dependent expertise and some sysadmin expertise.
PUDS	Power user Developing a Service. High degree of computing expertise.	As PUA/PUS but developing expertise like SP.
SP	Service Provider. High degree of computing expertise.	As PUA/PUS but has expertise in authorisation and possibly identity management.
Grid-Sys	Infrastructure sysadmin. High degree of computing expertise.	System administration of grid nodes, possibly with infrastructure delivery and security expertise.

'categorisation' activity, there will be users who move frequently between the groups, and whom may occupy two or more categories simultaneously. However, we believe that the categories are useful in examining high level requirements, especially those of access control and security.

Note that there are clearly omissions from Table 1. Two notable actors are the Third Party Beneficiary (TPB) and Resource Owner. A TBP could be a person or organisation who/which does not interact directly with the grid but whose personal data are being handled on the grid. Resource Owners clearly have important functions, but they do not necessarily interact with the grid, unless playing one of the seven main roles shown in Table 1 at a particular moment in time. In designing future grids, the requirements of both of these actors would have to be given much thought and would impact upon the likely architecture and security mechanisms of those grids. However, for the purposes of this paper, the general requirements (or expectations) of only the seven main roles are considered in relation to access management and other security needs.

Throughout this paper, the abbreviation SEU is taken to represent SEUD and SEUX where a statement could apply equally to either category.

On the ESP-GRID project wiki at http://wiki.oucs.ox.ac.uk/esp-grid/UserCategoryExampleActivities, we outline some example illustrations of these seven major actors. There was not room, in this brief paper, to describe them here. The majority of today's users come into the PUS and Grid-Sys categories (see Table 3, below).

Note that we have not divided the users into the kinds of grid jobs that result from their activity. This may, however, be another valuable approach. For example, one type of user may run a job that executes (or interacts with) only one grid node (a 'single point' job) whereas another may run a job that is divided, or subsequently splits, into many sub-jobs that interact with many grid nodes. These are useful definitions but are probably applicable to nearly all of the actors in Table 1.

3.2. Access management characteristics of these actors

Table 2 describes the access management or security characteristics of the seven user types. The final column of 'Security risk to grid node' tries to capture both the concepts of the threat to the grid resource and the risks (or costs) associated with managing these kinds of users. For example, the threat from an individual user of this type may be fairly low, but the difficulties of managing many users of this type give rise to an associated threat of attackers posing as those users. These are separate concepts, but they have been combined in this case, as it would appear to be appropriate.

The SEUD only ever uses a service, probably presented through some sort of gateway to the grid beyond. Therefore, the security risk to the grid resources from this user should be much lower than the other users. It is assumed that this user cannot interact directly with any grid nodes. Whatever threats that may exist from this type of user, there is the added defence of a restrictive application layer between the user and the grid node.

A similar profile could be expressed for the SEUX. However, a greater risk exists from those users due to executable code being run. Note also that there are similarities between the SEUX and PUA, using a resource broker. The main differences are in computing expertise, the use of a SP and that one is a true end user.

The PUA does not interact directly with any grid node (apart from the resource broker) and therefore should pose a lower

security risk than the other users, apart from the SEU. Nevertheless, code written by or submitted by the PUA will be run on a grid node somewhere and therefore the security risk may be seen as being moderate.

The PUS interacts directly with grid nodes, running code on those nodes. The security risk, from the viewpoint of the resource owners, is therefore much higher from this type of user and, if her identity is not known, it would be a requirement that

Table 2 Access management/security characteristics of the seven user types

User/actor	Access management/security characteristic	Security risk to grid node
SEUD	SEUD does not need to be 'known' by a grid access management service (should one exist) as the grid trusts and accounts the SP not the user. SP may need to authenticate, authorise and account for the user as well as possibly taking on 'metering' responsibilities.	Low (shielded by gateway/ application).
SEUX	The SEUX may have a similar access management characteristic to the SEUD due to the possible greater absolute numbers. The presence of SEUX will probably mandate the automated trace/isolate functionality discussed in section 4.3 on page 10.	Moderate.
PUA	The PUA's identity need not be managed by a grid access management service (should one exist) but some sort of mapping to a billing account may be necessary. It could be possible for the identity of the PUA to be concealed behind another entity, as occurs with the SEU. This entity could be a SP providing grid brokering services. Either the SP or the grid access management service is likely to require status (and other) information from an identity manager/provider for authorisation purposes.	Moderate (shielded by resource broker).
PUS	As for PUA, above in some scenarios. However, in addition, grid node owners may wish to have a direct authentication, authorisation (and accounting) relationship with the PUS. Alternatively, authentication elsewhere may be acceptable if a more transparent assertion of identity is given in order to satisfy the security instincts of grid node owners.	Moderate/high.
PUDS	As for PUS but moving into arrangements like SP (see below). May need to begin interacting with and accounting for SEUs in an experimental manner.	High (as for SP, see below).
SP	A SP may be trusted to provide services only to those authorised to use the grid or the SP may offer services to any end user, and be simply billed by the grid, or by the nodes that it uses. The SP may wish to manage identities and to authenticate SEUs or the SP may be willing to devolve these tasks. The SP probably needs to manage or recognise status (authorisation-related attributes). The SP needs strong/secure assertions of identity/authentication between it and the grid resource nodes. Accounting may be required between the grid resource nodes (or access management service) and the SP and between the SP and the SEU, although this latter requirement may not need to be met using grid middleware. As an individual, the SP could use any method (including that of devolved authentication) of access management to his/her machines (to which the SEUs connect or utilise in some way). Moreover, those machines may or may not be considered to be part of the grid.	High (impacts security both of grid nodes and of SEUs).
Grid-Sys	A Grid-Sys is likely to need to authenticate directly to particular grid resource nodes. However, in theory, it is possible that he may authenticate elsewhere and the node computer may trust that external authentication point (or identity provider).	Moderate (High risk but more easily managed).

it could be traced easily and very quickly, should any 'breach in security' occur. The benefits of a system whereby the user can be traced accurately, when problems occur, should outweigh the benefits (if there are any) of logging explicit identities at each grid node. See Norman (2006) for a further examination of the issues of asserting explicit identity 'up front'.

The PUDS has a similar security profile to the PUS, but is beginning to take on some of the aspects of a SP and therefore could pose a threat to both the grid and to the test SEUs involved in the development. When interacting with the grid, there may therefore be a requirement for the PUDS to be explicitly identified.

As Table 2 indicates, the SP has a complex security profile. The SP (machine) is likely to be trusted by and/or to be explicitly identified to both SEUs and to grid nodes. The SP (user) is also likely to have a profile similar to a PUDS when developing and testing and connecting to grid nodes directly. The SP machines may or may not be considered as part of the grid: these machines may simply be gateways to the grid and not contribute directly to grid computation.

The security profile of the Grid-Sys has been expressed as 'Moderate'. This is due to two opposing influences. Firstly, for each grid node, there will be very few system administrators, almost certainly in single figures. This means that the task of managing these users' authorisation information – and possibly authentication mechanisms – is relatively simple. Secondly and conversely, if an impostor were to be able to bypass the access management system, the risks are very high to the grid node.

3.3. An access management scenario

3.3.1. Two major routes of entry to the grid

Figure 1 outlines a likely scenario illustrating the access management 'behaviour' of these different types of grid users. As we have already established in Table 2, there are a variety of ways in which the access management requirements of each set of users, and of each resource protecting itself from each user, may be fulfilled. The scenario presented in Figure 1 is merely one of many that are possible. Nevertheless, we have depicted the PUA acting in two different ways, as these two ways are likely to be significant.

3.4. Proportions of users and our effort in servicing them

The assertions made in Table 3 are opinion only. However, if these are correct, then we need to find a way of engaging with users in the categories that are likely to account for a medium or high proportion of grid use in the future. It is obviously quite difficult to service such users when their current abundance is low and very tempting to over-engage with the current most common type of user.

4. The Customer-Service Provider grid relationship

4.1. SEUs dominate

It is clear, from our earlier assumptions, that the vast majority of 'users' of the grid, in future, will probably be Service End Users and, individually, these SEUs pose the lowest security threat as their activities are highly controlled by the SP and the service application. If we accept these as basic assumptions, we can see some advantages for the simplicity of a multi-tiered security architecture. As Figure 1 shows, there is trust between the grid and the SP and between the SP and the entity or organisation managing the user's credentials. Furthermore, the SP and the IdP are clear auditable points. We can thus envisage the SP as the true grid user. It is the SP entity that runs jobs on the grid

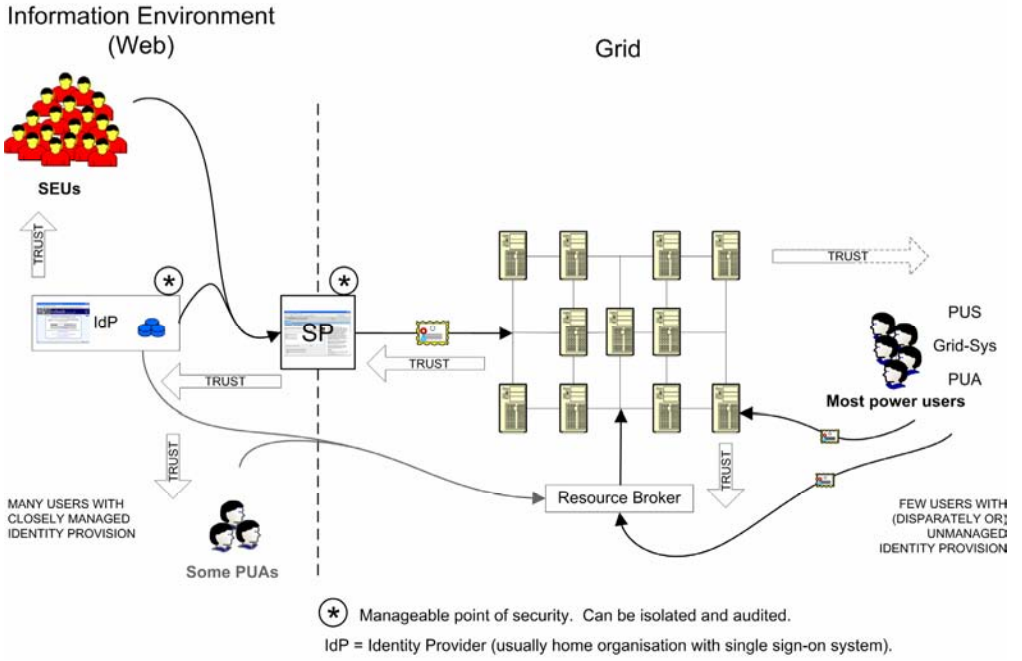


Figure 1 Possible access management behaviour scenario of the different types of grid users. (The PUDS has been omitted as it should contain elements of the PUS and SP).

and, if it were a commercial grid, the owners of the grid nodes could charge the SP for the use of their resource. Thus a clear relationship between the grid and the SP begins to emerge. Where particular authorisation requirements exist – such as "only members of organisation *A* can use this grid at this time" – the grid could mandate that the SP honour that requirement, and the SP could be audited for this. The SP is thus required to take responsibility for authorising users.

Table 3 Likely future proportions of grid users in each category

Type of user	Proportion of grid users by category	
	Current	Future
SEUD	Low	High
SEUX	Low	Medium
PUA	Medium	Low/medium
PUS	High	Low
PUDS	Low	Low
SP	Low/medium	Low
Grid-Sys	High	Low

This Customer-Service Provider concept does not need to rely upon any financial requirements. Even in an academic world – but one in which access is restricted to only certain communities – it would be appropriate to run application-based services to high numbers of users in this way.

For ease of use, the vast majority of users will access the power of grids via portals, portlets or similar server-based applications. If we accept that this is true, then we can take the opportunity to tighten up security for all of these users. The portal/server represents a point to which we can – technically or legally – devolve the responsibility for authentication and authorisation. This is a truly synergistic opportunity by:

- improving usability to users who would never benefit from the grid if it meant that they had to perform technical

computing operations to reach that point;

- introducing an 'auditable' point of security to which authentication and authorisation may be securely devolved.

The BRIDGES project built both a data and a compute grid infrastructure accessible by a portal which allowed biomedical researchers to authenticate (using a simple username/password mechanism)[1]. Scientists were then able to upload nucleotide (or protein) sequences and compare them against a variety of local and remote genomic databases. Explorations in rolling out X.509 *user* certificates to the BRIDGES scientists, for identity/authentication purposes, were largely unsuccessful. Instead, solutions utilising X.509 *server* certificates were adopted. Scientists were more comfortable with username/password solutions and to encourage uptake, these requirements were directly addressed. Numerous other challenges were tackled in BRIDGES such as re-engineering of client side tools for simplicity and user friendliness, e.g. to make them "google-like". In short, the scientists wanted a familiar environment in which to work, which shielded them as far as possible from the underlying Grid infrastructure.

Similarly, the Market for Computational Services project (Grid Markets, 2003) asserts that the evolution of the grid is constrained by the fact that users can only use machines where they have accounts. This approach is largely – but not entirely – aimed at Power Users in that the user has to engage at a much more technical level with each grid node. The user experience is greatly simplified in the Grid Markets project by interacting via a central broker. A logical extension to the findings of the BRIDGES and Grid Markets projects means that a lack of usability can mean an absolute lack of take-up, which in turn makes it difficult to survey users regarding their usability comments.

4.2. The main threat with the Customer-Service Provider model

The main threat with the Customer-Service Provider model, if implemented efficiently, is likely to be from denial of service (DoS) attacks on SPs. From the grid's or grid node's point of view, the user is the SP. Should any breach in security occur, the normal reaction would be to revoke the SP's privileges, temporarily or permanently. This seems reasonable. However, this means that all users benefiting from the service provided by the SP and the grid will by stymied.

A balance would need to be struck between the risk of this threat and the ability of the SPs to build reasonably safe applications. With such an application as the BLAST technology provided-for by the BRIDGES project, for example, it is difficult to see many threats to the SP other than:

- poor application/API control allowing (for example) SQL insert and update (etc.) statements;
- users submitting jobs incessantly, and thus tying up the databases and the compute cycles;
- submitting a cleverly formulated nucleotide sequence that never resolves and stays busy (as an extreme example).

Clearly, problems will occur, as they do with any multi-user application, but they should be able to be either mitigated-for in advance, or dealt with as they arise.

If a SP provides an application with very poor security then that SP clearly deserves to be suspended until such problems are fixed.

[1] See http://wiki.oucs.ox.ac.uk/esp-grid/NeSC_Shibbolized_Resources

4.3. Automated suspension

Rogue, clever, end users (or attackers who are, apparently, end users) will always exist and these need to be quickly identified. A high-level description of the need for automated suspension is discussed in Norman (2006). This could supersede the need for explicit identity assertion 'up front' and may make the C-SP model more secure.

5. Shibboleth

Building on the previous sections, we have established how the majority of users of a grid may be 'funnelled' via a server-based application so that requests and jobs may be run on the grid for them. Norman (2006) provides further details as to the case for using Shibboleth with grids and also points out some difficulties. However, the C-SP model would appear to make the use of Shibboleth more attractive.

6. Conclusions

The main conclusions of this hypothetical thinking regarding the likely users of future grids are:

- Like the mature web, we predict that most users will require simple, secure, ring-fenced applications to obtain the great benefits of grid technology.
- If such applications are placed in portals (probably using web technology), the security threat profile of this vast majority of users is relatively low (being partly mediated by the application). Thus, heavyweight security solutions will not be needed for the majority of users.
- In such a scenario, Power Users will exist as a small proportion of users. Those users probably merit heavyweight security solutions to be applied to them.
- Where applications are based for the benefit of most users, these provide convenient 'funnels' for such users. Such funnels are suitable for security auditing and therefore are a substantial aid to scalability.
- We have categorised the majority of users as Service End Users (SEUs) who interact directly with Service Providers (SPs). In this Customer-Service Provider model, it is the SPs that interact with the grid directly.
- Grids will be used by many Power Users. We have tentatively named these as PUAs, PUSs, PUDSs, (SPs) and Grid-Sys's. There are more 'actors' in such a system, but we believe that these capture most of the users who interact with the grid directly.
- We described the concept of the SEU-SP interaction as the 'Customer-Service Model'.

7. References

Beckles, B. (2004a) Removing digital certificates from the end-user's experience of grid environments. UK eScience All Hands Meeting (2004) http://www.allhands.org.uk/2004/proceedings/papers/250.pdf.

Beckles, B. (2004b) User requirements for UK e-Science grid environments. UK e-Science All Hands Meeting (2004) http://www.allhands.org.uk/2004/proceedings/papers/251.pdf.

Grid Markets (2003). A Market for Computational Services: A Proposal to the e-Science Core Technology Programme. http://www.lesc.ic.ac.uk/markets/Resources/Tag.pdf . Also http://www.sve.man.ac.uk/Research/AtoZ/MCS/RUS/.

Norman, M.D.P. (2006) A case for Shibboleth and grid security: are we paranoid about identity? Proceedings of the 2006 UK e-Science All Hands Meeting

Sinnott, R (2006) Development of Usable Grid Services for the Biomedical Community. Proceedings of *Designing for e-Science: Interrogating new scientific practice for usability, in the lab and beyond*' workshop at the UK National e-Science Centre, January 25-26, 2006.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Martlet: A Scientific Work-Flow Language for Abstracted Parallisation

Daniel Goodman
Oxford University Computing Laboratory, Parks Road, Oxford, OX1 3QD, UK
Daniel.Goodman@comlab.ox.ac.uk

27th July 2006

Abstract

This paper describes a work-flow language 'Martlet' for the analysis of large quantities of distributed data. This work-flow language is fundamentally different to other languages as it implements a new programming model. Inspired by inductive constructs of functional programming this programming model allows it to abstract the complexities of data and processing distribution. This means the user is not required to have any knowledge of the underlying architecture or how to write distributed programs.

As well as making distributed resources available to more people, this abstraction also reduces the potential for errors when writing distributed programs. While this abstraction places some restrictions on the user, it is descriptive enough to describe a large class of problems, including algorithms for solving Singular Value Decompositions and Least Squares problems. Currently this language runs on a stand-alone middleware. This middleware can however be adapted to run on top of a wide range of existing work-flow engines through the use of JIT compilers capable of producing other work-flow languages at run time. This makes this work applicable to a huge range of computing projects.

1 Introduction

The work-fbw language Martlet described in this paper implements a new programming model that allows users to write parallel programs and analyse distributed data without having to be aware of the details of the parallelisation. It abstracts the parallelisation of the computation and the splitting of the data through the inclusion of constructs inspired by functional programming. These allow programs to be written as an abstract description that can be adjusted to match the data set and available resources automatically at runtime. While this programming model adds some restriction to the way programs can be written, it is possible to perform complex calculations across a distributed data set such as Singular Value Decomposition or Least Squares problems, and it creates a much more intuitive way of working with distributed systems. This allows inexperienced users to take advantage of the power of distributed computing resources, and reduces the work load on experienced distributed programmers.

While applicable to a wide range of projects, this was originally created in response to some of the problems faced in the distributed analysis of data generated by the Climate*Prediction*.net[1][9, 12] project. Climate*Prediction*.net is a distributed computing project inspired by the success of the SETI@home[2][1] project. Users download a model of the earth's climate and run it for approximately fifty model years with a range of perturbed control parameters before returning results read from their model to one of the many upload servers.

The output of these models creates a data set that is distributed across many servers in a well-defined fashion. This data set is too big to transport to a single location for analysis, so it must be worked on in a distributed manner if a user wants to analyse more than a small subset of the data. In order to derive results, it is intended that users will submit analysis functions to the servers holding the data set. As this data set provides a resource for many people, it would be unwise to allow users to submit arbitrary source code to be executed. In addition users are unable to ascertain how many servers a given subset of this data that they want to analyse spans, and nor should they care. Their interest is in the information they can derive from the data, not how it is stored. These requirements mean a trusted work-fbw lan-

[1] http://www.climateprediction.net
[2] http://setiathome.ssl.berkeley.edu

guage is required as an intermediate step, allowing the construction of analysis functions from existing components, and abstracting the distribution of the data from the user.

2 Related Work

Existing work-flow languages such as BPEL[2], Pegasus [7] and Taverna [11] allow the chaining together of computational functions to provide additional functions. They have a variety of supporting tools and are compatible with a wide range of different middlewares, databases and scientific equipment. They all implement the same programming model where a known number of data inputs are mapped to computational resources and executed, taking advantage of the potential for parallelisation where possible and supporting *if* and *while* statements *etc*. As they only take a known number of inputs, none of them are able to describe a generic work-flow in which the number of inputs is unknown, which the middleware can then adapt to perform the described function at runtime once the number of inputs is known.

Independently Google have developed a programming model called Map-Reduce [6] to perform distributed calculations. This is similar to, but not as general, or as loosely coupled as Martlet. The implementing library works with the Google File System [8] to allow parallel calculations on data, while abstracting the complexity of the data storage and processing. Though similar, as it is aimed at the internal work of Google programmers working with large clusters. As such it is a set of objects that are dependant on the Google infrastructure and extended by the user. These require that the user to provide information about the environment such as the number of computers to be involved in the calculation, and functions to partition the data. All of these make it not suited to the more public heterogeneous domain that this project is aimed at.

3 Example Problem

The average temperature of a given set of returned models is an example of a situation where the level of abstraction described in this paper is required. If this data spans a servers, this calculation can be described in way that could be used for distributed computing as:

$$y_0 = \sum_{i=0}^{n_1-1} x_i$$
$$z_0 = n_1$$

$$y_1 = \sum_{i=n_1}^{n_2-1} x_i$$
$$z_1 = n_2 - n_1$$

$$\vdots$$
$$y_{a-1} = \sum_{i=n_{a-1}}^{n_a-1} x_i$$
$$z_{a-1} = n_a - n_{a-1}$$

$$\overline{x} = \frac{\sum_{i=0}^{a-1} y_i}{\sum_{i=0}^{a-1} z_i}$$

where each subset of the data set has a computation performed on it, with the results used by a final computation to produce the over all average. Each of these computations could occur on a different computing resource.

To write this in an existing work-flow language in such a way that it is properly executed in parallel, the user must first find out how many servers their required subset of data spans. Only once this value is known can the work-flow be written, and if the value of a changes the work-flow must be rewritten. The only alternative is that the user himself must write the code to deal with the segregated data. It is not a good idea to ask this of the user since it adds complexity to the system that the user does not want and may not be able to deal with, as well as adding a much greater potential for the insertion of errors into the process. In addition, work-flow languages are not usually sufficiently descriptive for a user to be able to describe what to do with an unknown number of inputs, so it is not possible just to produce a library for most languages This problem is removed with Martlet, by making such abstractions a fundamental part of the language.

4 Introducing Martlet

Our work-flow language *Martlet* supports most of the common constructs of the existing work-flow languages. In addition to these constructs, it also has constructs inspired by inductive constructs of functional programming languages [5]. These are used to implement a new programming model where functions are submitted in an abstract form and are only converted into a concrete function that can be executed when provided with concrete data structures at runtime. This hides from the user the parallel nature of the execution and the distribution of the data they wish to analyse.

We chose to design a new language rather than extending an existing one because the widely used languages are already sufficiently complex that an extension for our purposes would quickly obfuscate the features we are aiming to explore. Moreover, at the time the decision was taken, there were no suitable open-source work-flow language implementations to adapt. It is hoped that in due course the ideas developed in this language will be added into other languages.

The inspiration for this programming model came from functional programming languages where it is possible to write extremely concise powerful functions based on recursion. The reverse of a list of elements for instance can be defined in Haskell [5] as;

```
reverse [] = []
reverse (x:xs) = reverse xs ++ [x]
```

This simply states that if the list is empty, the function will return an empty list, otherwise it will take the first element from the list and turn it into a singleton list. Then it will recursively call reverse on the rest of the list and concatenate the two lists back together. The importance of this example is the explicit separation between the base case and the inductive case. Using these ideas it has been possible to construct a programming model and language that abstracts the level of parallelisation of the data away from the user, leaving the user to define algorithms in terms of a base case and an inductive case.

Along with the use of functional programming constructs, two classes of data structure, *local* and *distributed*, were created. Local data structures are stored in a single piece; distributed data structures are stored in an unknown number of pieces spanning an unknown number of machines. Distributed data structures can be considered as a list of references to local data structures. These data structures allow the functional constructs to take a set of distributed and local data structures, and functions that perform operations on local data structures. These are then used as a base case and an inductive case to construct a work-flow where the base function gets applied to all the local data structures referenced in the distributed data structures, before the inductive function is used to reduce these partial results to a single result. So, for example, the distributed average problem looked at in Section 3, taking the distributed matrix `A` and returning the average in a column vector `B`, could be written in Martlet as the program in Figure 1.

Due to this language being developed for large scale distributed computing on huge data sets, the data is passed by reference. In addition to data, functions are also passed by reference. This means that functions are first class values that can be passed into and used in other functions, allowing the workflows to be more generic.

5 Syntax and Semantics

To allow the global referencing of data and functions, both are referenced by URIs. The inclusion of these in scripts would make them very hard to read and would increase the potential for user errors. These problems are overcome using two techniques. First, local names for variables in the procedure are used, so the URIs for data only need to be entered when the procedure is invoked. This means that in the procedure itself all variable names are short, and can be made relevant to the data they represent. Second, a define block is included at the top of each procedure where the programmer can add abbreviations for parts of the URI. This works because the URIs have a logical pattern set by whom the function or data belongs to and the server it exists on. As a result the URIs in a given process are likely to have much in common.

The description of the process itself starts with the keyword "`proc`", then there is a list of arguments that are passed to the procedure, of which there must be at least one due to the stateless nature of processes. While additional syntax describing the read, write nature of the arguments could improve readability, it is not included as it would also prevent certain patterns of use. This may change in future variants of the language. Finally there is a list of statements in between a pair of curly braces, much like C. These statements are executed sequentially when the program is run.

There are two types of statement: normal statements and expandable statements. The difference between the two types of statements is the way they behave when the process is executed. At runtime an *expand* call is made to the data structure representing the abstract syntax tree. This call makes it adjust its shape to suit the set of concrete data references it has been passed. Normal statements only propagate the *expand* call through to any children they have, whereas expandable statements adjust the structure of the tree to match the specific data set it is required to operate on.

5.1 Normal Statements

As the language currently stands, there are six different types of normal statement. These are if-else, sequential composition, asynchronous composition, while, temporary variable creation, and process calls. Their syntax is as follows:

Sequential Composition is marked by the keyword `seq` signalling the start of a list of statements that need to be called sequentially. Although the `seq` keyword can be used at any point where a statement would be expected, in most places sequential composition is implicit. The only location that this construct really is required is when one wants to create a function in which a set of sequential lists

```
// Declare URI abbreviations in order to improve the script readability
define
{
   uri1 = baseFunction:system:http://cpdn.net:8080/Martlet;
}

proc(A,B)
{
// Declare the required local variables for the computation. Y and Z
// are used to represent the two sets of values Yi and Zi in the
// example equations. ZTotal will hold the sum of all the Zi's.
   Y = new dismatrix(A);
   Z = new disinteger(A);
   ZTotal = new integer(B);

// The base case where each Yi and Zi is calculated, and recorded in
// Y and Z respectively. The map construct results in each Zi and Yi
// being calculated independently and in parallel.
   map
   {
      matrixSum:uri1(A,Y);
      matrixCardinality:uri1(A,Z);
   }

// The inductive case, where we sum together the distributed Yi's
// and Zi's into B and ZTotal respectively.
   tree((YL,YR)\Y -> B, (ZL,ZR)\Z -> ZTotal)
   {
      matrixSumToVector:uri1(YL,YR,B);
      IntegerSum:uri1(ZL,ZR,ZTotal);
   }
// Finally we divide through B with ZTotal to finish computing the
// average of A storing the result in B.
   matrixDivide:uri1(B,ZTotal,B);
}
```

Figure 1: Function for computing the average of a matrix *A* split across an unknown number of servers. The syntax and semantics of this function is explained in Section 5.

of statements were run concurrently by an asynchronous composition. An example of this is shown in Figure 2

Asynchronous Composition is marked by the keyword `async` and encompasses a set of statements. When this is executed each statement in the set is started concurrently. The asynchronous statement only terminates when all the sub-statements have returned.

In order to prevent race conditions it is necessary that no process uses a variable concurrently with a process that writes to the variable. This is enforced by the middleware at runtime.

if-else & while are represented and behave the same as they would in any other procedural language. There is a test and then a list of statements.

Temporary Variables can be created by statements that look like

```
identifier =
        new type(identifier);
```

The identifier on the left hand side of the equality is the name of the new variable. The type on the right is the type of the variable, and the identifier on the right is a currently existing data structure used to determine the level of parallelisation required for the new variable. For example if the statement was

```
A = new DisMatrix(B);
```

this will create a distributed matrix `A` that is split into the same number of pieces as `B`. The type field is required as there is no constraint that the type of `A` is the same as the type of `B`. This freedom is required as there is no guarantee that a distributed data structure of the right type is going to appear at this stage in the procedure, as was the case in the average calculation example in Figure 1.

Process calls fall into one of two categories. They can either be statically named in the function or passed in as a reference at runtime. Both appear as an identifier and a list of arguments.

5.2 Expandable Statements

The are four expandable statements, `map`, `foldr`, `foldl` and `tree`. Each of these has a functional programming equivalent. Expandable statements don't propagate the call to expand to their children and must have been expanded before the function can be computed. This means that on any given path between the root and a leaf there must be at most one expandable statement.

```
async{
      seq{
          function1(A,B,C);
          function2(A,B);
          function3(B,C);
      }

      seq{
          function4(D,E);
          function1(D,E,F);
          function5(E,F);
      }
}
```

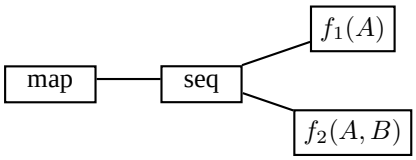
Figure 2: `seq` used to run two sequential sets of operations asynchronously.

map is equivalent to `map` in functional programming where it takes a function `f` and a list, and applies this function to every element in the list. This is shown below in Haskell:

```
map f [] = []
map f (x:xs) = (f x):(map f xs)
```

Map in Martlet encompasses a list of statements as shown in the example below. Here function calls `f1` and `f2` are implicitly joined in a sequential composition to create the function represented by `f` in the Haskell definition. The list is created by distributed values `A` and `B`. While in its unexpanded abstract form, this example maps onto the abstract syntax tree also shown below.

```
map
{
  f1(A);
  f2(A,B);
}
```



When this is expanded, it looks at the distributed data structures it has been passed and creates a copy of these statements to run independently on each piece of the distributed data structure as shown in Figure 3.

Due to the use of an asynchronous statement in this transformation, no local value that is passed into the map statement can be written to. However local values created within the map node can be written to.

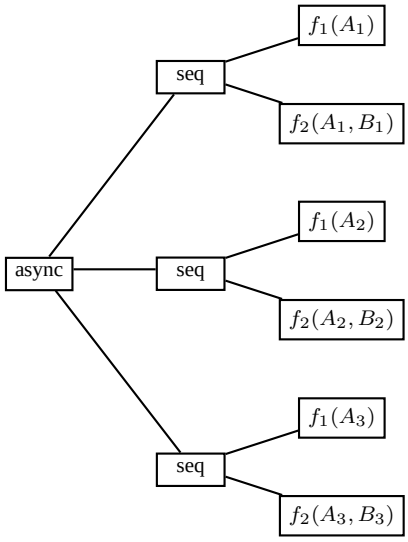


Figure 3: The abstract syntax tree for the example map statement after expand has been called setting $A = [A_1, A_2, A_3]$ and $B = [B_1, B_2, B_3]$.

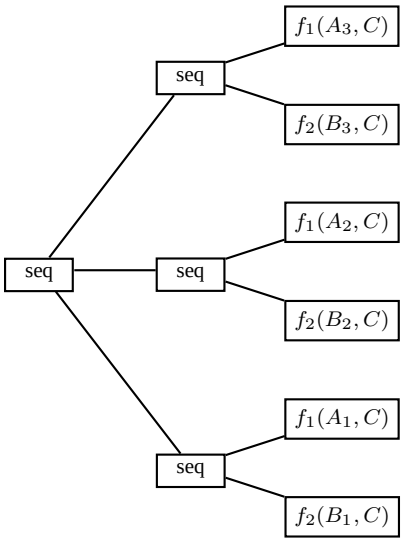


Figure 4: The abstract syntax tree for the example foldr statement after expand has been called setting $A = [A_1, A_2, A_3]$ and $B = [B_1, B_2, B_3]$.

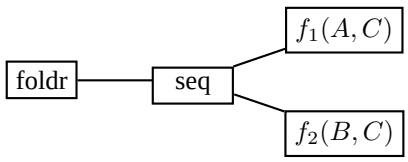
foldr is a way of applying a function and an accumulator value to each element of a list. This is defined in Haskell as:

```
foldr f e [] = e
foldr f e (x:xs) = f x
                   (foldr f e xs)
```

This means that the elements of a list `xs = [1,2,3,4,5]` can be summed by the statement; `foldr (+) 0 xs` which evaluates to `1+(2+(3+(4+(5+0))))`

Foldr statements are constructed from the `foldr` keyword followed by a list of one or more statements which represent `f`. An example is shown below with its corresponding abstract syntax tree.

```
foldr
{
   f1(A,C);
   f2(B,C);
}
```



When this function is expanded this is replaced by a sequential statement that keeps any non-distributed arguments constant and calls `f` repeatedly on each piece of the distributed arguments as shown in Figure 4.

foldl is the mirror image of foldr so the Haskell example would now evaluate to `((((0+1)+2)+3)+4)+5`

The syntax tree in Martlet is expanded in almost exactly the same way as foldr. The only difference is the function calls from the sequential statement are in reverse order. The only time that there is any reason to choose between foldl and foldr is when `f` is not commutative.

tree is a more complex statement type. It constructs a binary tree with a part of the distributed data structure at each leaf, and the function `f` at each node. When executed this is able to take advantage of the potential for parallel computation. A Haskell equivalent is:

```
tree f [x] = x
tree f (x:y:ys) =
     f (tree f xs') (tree f ys')
     where (xs',ys') =
                  split (x:y:ys)
```

`split` is not defined here since the shape of the tree is not part of the specification. It will however always split the list so that neither is empty.

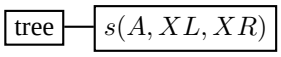
Unlike the other expandable statements, each node in a tree takes $2n$ inputs from n distributed data structures, and produce n outputs. As there is insufficient information in the structure to construct the mappings of values between nodes within the tree, the syntax requires the arguments that the statements use to be declared in brackets above the function in

such a way that the additional information is provided.

Non-distributed constants and processes used in `f` are simply denoted as a variable name. The relationship between distributed inputs and the outputs of `f` are encoded as `(ALeft,ARight)\A->B`, where `ALeft` and `ARight` are two arguments drawn from the distributed input `A` that `f` will use as input. The output will then be placed in `B` and can be used as an input from `A` at the next level in the tree.

Lets consider a function that uses a method sum passed into the statement as `s`, a distributed argument `X` as input and outputs the result to the non-distributed argument `A`. This could be written as:

```
tree((XL,XR)\X -> A)
{
   s(A,XL,XR);
}
```



When this is expanded, it uses sequential, asynchronous and temporary variables in order to construct the tree as shown in Figure 5. Because of the use of asynchronous statements any value that is written to must be passed in as either an input or an output.

5.3 Example

If the Martlet program to calculate averages from the example in Figure 1 where submitted it would produce the abstract syntax tree shown in Figure 6. This could then be expanded using the techniques show here to produce a concrete functions for different concrete datasets.

6 Conclusions

In this paper we have introduced a language and programming model that use functional constructs and two classes of data structure. Using these constructs it is able to abstract from users the complexity of creating parallel processes over distributed data and computing resources. This allows the user simply to think about the functions they want to perform and does not require them to worry about the implementation details.

Using this language, it has been possible to describe a wide range of algorithms, including algorithms for performing Singular Value Decomposition, North Atlantic Oscillation and Least Squares.

To allow the evaluation of this language and programming model, a supporting middleware has been constructed [10] using web services supported by Apache Axis [3] and Jakarta Tomcat [4]. As we have found no projects with a similar approach aimed at a similar style of environment, a direct comparison with other projects has not been possible. This work is, however, currently being tested with data from the Climate*Prediction*.net project with favorable results and will hopefully be deployed on all our servers over the course of the next year allowing testing on a huge data set.

At runtime, when concrete values have been provided, it is possible to convert abstract functions into concrete functions. The concrete functions then contain no operations that are not supported by a range of other languages. As such, it is envisaged that the middleware will in time be cut back to a layer that can sit on top of existing work-flow engines, providing extended functionality to a wide range of distributed computing applications. This capability will be provided through the construction of a set of JIT compilers for different work-flow languages. Such compliers need only take a standard XML output produced at runtime and performing a transformation to produce the language of choice. This would then allow a layer supporting the construction of distributed data structures and the submission of abstract functions to be placed on top of a wide range of existing resources with minimal effort, extending their use without affecting their existing functionality. Such a middleware would dramatically increase the number of projects that Martlet is applicable to. Hopefully the ideas in Martlet will then be absorbed into the next generation of work-flow languages. This will allow both existing and future languages to deal with a type of problem that thus far has not been addressed, but will become ever more common as we generate ever-larger data sets.

References

[1] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. Seti@home: an experiment in public-resource computing. *Commun. ACM*, 45(11):56–61, 2002.

[2] Tony Andrews, Francisco Curbera, Hitesh Doholakia, Yaron Goland, Johannes Kiein, Frank Leymann, Kevin Liu, Dieter Roller, Doug Smitth, Satish Thatte, Ivana Trickovic, and Sanjiva Weerwarana. BPEL4WS. Technical report, BEA Systems, IBM, Microsoft, SAP AG and Siebel Systems, 2003.

[3] Apache Software Foundation. *Apache Axis*, 2005. URL: http://ws.apache.org/axis/.

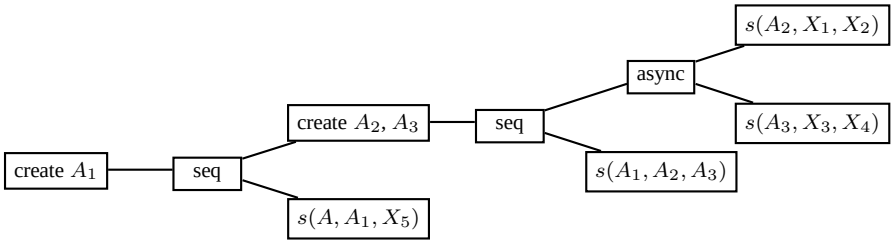


Figure 5: When the tree function on page 7 is expanded with $X = [X_1, X_2, X_3, X_4, X_5]$, this is one of the possible trees that could be generated.

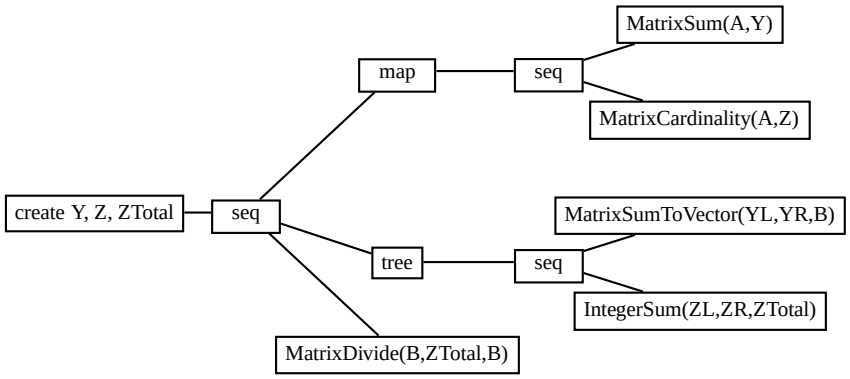


Figure 6: The abstract syntax tree representing the generic work-flow to compute the an average introduced in Figure 1.

[4] Apache Software Foundation. *The Apache Jakarta Project*, 2005. URL: http://jakarta.apache.org/tomcat/.

[5] Richard Bird. *Introduction to Functional Programming using Haskell*. Prentice Hall, second edition, 1998.

[6] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. Technical report, Google Inc, December 2004.

[7] E. Deelman, J. Blythe, Y. Gil, and C. Kesselman. Pegasus: Planning for execution in grids. Technical report, Information Sciences Institute, 2002.

[8] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 29–43, New York, NY, USA, 2003. ACM Press.

[9] Daniel Goodman and Andrew Martin. Grid style web services for climate*prediction*.net. In Steven Newhouse and Savas Parastatidis, editors, *GGF workshop on building Service-Based Grids*. Global Grid Forum, 2004.

[10] Daniel Goodman and Andrew Martin. Scientific middleware for abstracted parallelisation. Technical Report RR-05-07, Oxford University Computing Lab, November 2005.

[11] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R. Pocock, Anil Wipat, and Peter Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.

[12] David Stainforth, Jamie Kettleborough, Andrew Martin, Andrew Simpson, Richard Gillis, Ali Akkas, Richard Gault, Mat Collins, David Gavaghan, and Myles Allen. Climate*prediction*.net: Design principles for public-resource modeling research. In *14th IASTED International Conference Parallel and Distributed Computing and Systems*, Nov 2002.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

An Intelligent and Adaptable Grid-based Flood Monitoring and Warning System

Danny Hughes[1], Phil Greenwood[1], Gordon Blair[1], Geoff Coulson[1],

Florian Pappenberger[2], Paul Smith[2] and Keith Beven[2]

[1]Computing Department, Infolab21, Lancaster University, UK, LA1 4WA.

[2] Environmental Science Department, Lancaster University, UK, LA1 4YQ.

Abstract

Flooding is a growing problem in the UK. It has a significant effect on residents, businesses and commuters in flood-prone areas. The cost of damage caused by flooding correlates closely with the warning time given before a flood event, and this makes flood monitoring and prediction critical to minimizing the cost of flood damage. This paper describes a wireless sensor network for flood warning which is not only capable of integrating with remote fixed-network grids for computationally-intensive flood modeling purposes, but is also capable of performing on-site flood modeling by organising itself as a 'local grid'. The combination of these two modes of grid computation—local and remote—yields significant benefits. For example, local computation can be used to provide timely warnings to local stakeholders, and a combination of local and remote computation can inform adaptation of the sensor network to maintain optimal performance in changing environmental conditions.

1. Introduction

Flooding is a growing problem in the UK and affects a large number of people financially, physically and emotionally. The problem was dramatically highlighted by the wide-spread floods of Autumn 2000, the total cost of which was estimated to be in the order of £1 billion. Following these floods, major initiatives [Environment Agency '00] have been undertaken to improve the UK's flood readiness. These include: improving flood defenses; raising public awareness; and, significantly for this project, improving flood warning systems.

Traditionally, hydrologists have approached flood prediction by deploying sensors (such as depth and flow-rate sensors) at sites prone to flooding. Data from these sensors is then collected manually or via GSM-based telemetry and used as the input to flood prediction models. Two main classes of flood prediction model are commonly used. The first, referred to as *spatial* models [Pappenberger '05], provide detailed, site-wide predictions, albeit with limited accuracy at any given point. They are computationally complex and must be executed on clusters or grids. The second class, referred to as *point prediction* models [Beven '05], provide accurate depth predictions for a single point in the flood plain. They are computationally simple so that they may be executed in a timely fashion on standard desktop PC hardware.

In summary then, traditional flood monitoring approaches impose a rigid separation between the on-site wireless sensor networks (WSNs) that are used to collect data, and the off-site computational grid which is used to analyze this data. Essentially, the sensor networks are computationally 'dumb', being composed of nodes that are capable only of recording and transmitting sensor data.

In order to better support timely flood warnings, we argue that more on-site 'intelligence' is required. The 'GridStix' sensor platform presented in this paper uses powerful embedded hardware, heterogeneous wireless networking technologies, and next generation grid middleware to implement an adaptable WSN that doubles as a lightweight grid, allowing nodes to not only ship data to remote fixed grids, but also to perform 'local' grid computations with significant benefits as discussed below.

This paper describes the operation of a GridStix-based flood monitoring system and specifically focuses on how local grid computation can be used to support the adaptation of the WSN to changing environmental conditions. The remainder of this paper is structured as follows. First, section 2 discusses how local computation can be exploited in WSNs. Section 3 then introduces our 'GridStix' platform, section 4 highlights potential forms of adaptation that are available

using that platform, and section 5 discusses factors that can be used to drive adaptation. Finally, section 6 discusses our ongoing deployment and evaluation work, section 7 discusses related work, and section 8 offers conclusions and outlines our plans for future work.

2. Exploiting Local Computation

Our prototype flood prediction system uses local grid computation to provide improved support for flood monitoring. This section discusses how local computation can be used to (i) inform system adaptation, (ii) support diverse sensors and (iii) provide timely warnings to local stakeholders.

First, local computation can be used to drive the ***adaptation of WSN behaviour*** based on awareness of environmental conditions such as flood data and power monitoring. For example, based on the execution of point prediction models, we can switch to a more reliable network topology (see below) at times when node failure seems more likely (i.e. when imminent flooding is predicted). Adaptation may also be informed by input from computationally intensive spatial prediction models executed in the remote fixed grid.

Second, the availability of local computation can support ***richer sensor modalities*** such as image-based flow prediction [Bradley '03]. Image-based flow prediction is a novel technique for measuring water flow rates that uses off-the-shelf digital cameras. It is cheaper and more convenient to deploy than the commonly-used ultrasound flow sensors, but can only be used where significant computational power is available. Flow-rate measurements are calculated based upon a series of images taken by a digital camera deployed overlooking the river. Naturally occurring tracer particles are identified on the water surface and tracked through subsequent images, from which the surface velocity of the water is inferred. The data-set used by this method, a sequence of high-resolution images, is too large for off-site transmission to be feasible using GSM or GPRS technologies and therefore the method is impractical in current sensor network deployments. However, organising computationally-capable sensor nodes into a local grid allows analysis to be performed on-site and the results of this analysis then transmitted off-site.

Finally, on site flood-modeling allows ***timely flood warnings*** to be distributed to local stakeholders. These flood-warnings are based on the results of point prediction models executed by the local grid of GridStix nodes and disseminated to local stakeholders in a range of formats including on-site audio/visual warnings, a public web-site and SMS alerts. Each of these media has associated benefits and drawbacks. For example, SMS warnings are an effective method of publishing timely alerts to local stakeholders. However, SMS warnings require that users register for the service in advance and are therefore ineffective for stakeholders who might be unaware of a flood risk. Local audio/visual flood warnings may be effective without the need for stakeholders to proactively participate, however their effectiveness is dependent upon stakeholders being within audio/visual range.

3. The GridStix Platform

3.1 Overview

In order to achieve the 'local grid' functionality discussed in the previous section, a powerful and versatile sensor platform is required (in terms of both hardware and software). This section describes such a platform – GridStix. More information on the platform itself is given in [Hughes '06].

3.2 Hardware Platform

In order to support the proposed functionality, a sensor node device must be capable of interfacing with a variety of sensors including traditional sensors (e.g. depth sensors) and more novel sensors (e.g. the digital imaging hardware that is used to support the image-based flow prediction discussed above). A suitable device must also be capable of supporting a variety of wireless communications technologies to provide redundancy and allow sensor nodes to switch physical network technologies as conditions require. Finally, the device must have sufficient computational and storage resources to support the GridKit software platform (see below).

Sensor networks often make use of devices with extremely constrained local resources such as the Berkley Motes [Xbow '06]. This is because such devices have extremely modest power requirements and can therefore operate for long periods on small batteries. However, such constrained platforms do not offer sufficient computational power to support functionality such as on-site flood prediction, nor do they offer sufficient support for diverse networking technologies and sensor types. For this reason, more powerful embedded hardware

has been selected for use in the GridStix platform.

Each GridStix node is based on the *Gumstix* [Waysmall '06] embedded computing platform, so named as each device is roughly the same size as a pack of gum. Despite their small-size, each of these devices is equipped with a 400 MHz Intel XScale PXA255 CPU, 64Mb of RAM and 16Mb of flash memory. These hardware resources support the execution of a standard Linux kernel and Java Virtual Machine making them inherently compatible with the GridKit platform, which has been successfully deployed on comparable hardware such as PDAs [Cooper '05]. Furthermore, the PXA255, which performs comparably to a 266MHz Pentium-class CPU are capable of executing a single iteration of a point prediction model in a matter of seconds.

The Gumstix devices also provide a variety of hardware I/O mechanisms, enabling connection to a variety of sensors. For example, a network camera (for image-based flow measurement) can be connected via a standard wired Ethernet connection, while flow sensors can be connected via an on-board serial port, and depth sensors can be connected via the GPIO lines of the XScale I2C bus. In this way it is possible to connect multiple sensors to a single device. In terms of networking, each device is equipped with an onboard Bluetooth radio and Compact Flash 802.11b network hardware, which is used to provide an ad-hoc communications infrastructure. Furthermore, the devices can be equipped with GPRS modems for transmitting and receiving data from off-site.

Of course, the above capabilities come at the expense of increased power consumption: While a Berkley Mica Mote consumes only 54mW during active operation [XBow'06], our devices consume around 1W during typical operation, and thus it would not be feasible to power them for long periods using batteries alone. To address this, solar panel arrays are employed. Given aggressive power management, we have found that a single 15cm^2 mono-crystalline solar panel, with a maximum output of 1.9 watts combined with a 6v 10AH battery, is sufficient to continually power a device.

Finally, to minimise the effects of harsh weather conditions, flood water, vandalism, uncooperative grazing animals, etc., we have housed the devices in durable, water-tight containers that can safely be buried. In some cases burial is not possible (e.g. solar panel deployment). In such cases, the device is situated as discreetly and securely as possible to avoid unwanted attention.

3.3 The GridKit Middleware Platform

The GridKit middleware platform [Coulson '05] provides the key functionality that is required to support distributed systems such as grids, peer-to-peer networks and WSNs. GridKit is based on the OpenCOM [Coulson '02] component model and the various facets of system functionality are implemented as independent component frameworks. This component-based approach allows developers to build rich support for distributed systems or conversely, to build stripped-down deployments suitable for execution on embedded hardware such as the Gumstix.

Importantly, GridKit offers rich support for *application-level overlay networks*. Its Overlay Framework [Coulson '05] supports the simultaneous deployment of multiple overlay networks and enables these to interoperate flexibly (e.g. by layering them vertically or composing them horizontally). It also supports the adaptation of overlays, allowing for example, one overlay to be swapped for another at run-time. The use of adaptable overlays is discussed in detail in section 4, which illustrates how overlays with different performance characteristics can be used to adapt to changing environmental conditions.

4. Supporting Adaptation

4.1 Overview

This section examines situations in which WSN adaptation is possible and then goes on to consider the factors that can be used to inform such adaptation. Three discrete classes of adaptation are identified: (i) adaptation at the level of the physical network, (ii) adaptation at the overlay network level, and (iii) adaptation of CPU performance.

4.2 Physical Network Adaptation

Our flood prediction system makes use of three wireless networking technologies, Bluetooth, IEEE 802.11b and GPRS, each of which has very different performance characteristics:

- The compact flash 802.11b hardware (SanDisk Connect Plus) supports speeds of 11Mbps at a range of 137 meters, 5.5Mbps at 182 meters, 2 Mbps at 243 meters and 1Mbps at 365 meters. It offers significantly better performance than Bluetooth or GPRS

and has a maximum power consumption of approximately 0.5 watts.

- The class 2 Bluetooth radio (Ericsson ROK-104-001) supports speeds of up to 768Kbps at a range of up to 25M. It offers QoS that is significantly lower than 802.11b, but significantly higher than GPRS. It consumes a maximum of 0.2 watts [Ericsson '02].
- The GPRS modem (Ambicom GPRS-CF) supports uplink speeds of up to 29kbps and downlink speeds of up to 58kbps. Range is not an issue as we assume that the entire deployment site is within the bounds of GPRS coverage. GPRS offers lower QoS than either 802.11b or Bluetooth and consumes a maximum of 2 watts. Its operating wavelength of around 30cm is longer than that of the other technologies (which operate at around 12cm) and therefore performs better under water [Ambicom '06].

Each of our three communication technologies clearly has advantages and disadvantages. For example, 802.11b offers good QoS and long range; however, it consumes significantly more power than Bluetooth. Conversely GPRS offers much poorer performance, but is not limited by range. We discuss below how these differing characteristics can best be exploited.

4.3 Application-level Overlay Adaptation

As previously discussed, we employ application-level overlays to provide communications support for our flood prediction system. There are a range of overlays which can be used and each has advantages and disadvantages.

Off-site data dissemination is supported by the use of *spanning tree-based overlays*. These are commonly used in WSNs to disseminate data from a large number of sensors to a small number of logging or bridging nodes which form the 'root' of the tree. Prime examples of spanning trees are Shortest Path (SP) and Fewest Hop (FH) trees. FH trees are optimised to maintain a minimum of hopes between each node and the root. They minimise the data loss that occurs due to node failure, but are sub-optimal with respect to power consumption. SP trees, on the other hand, are optimised to maintain a minimum distance in edge weights from each node to the root. As a result, they tend to consume less power than FW trees, but are more vulnerable to node failure. Both forms of tree can be efficiently created using Dijkstra's algorithm [Dijkstra '59]. Examples of SP and FH spanning trees are shown in figure 1.

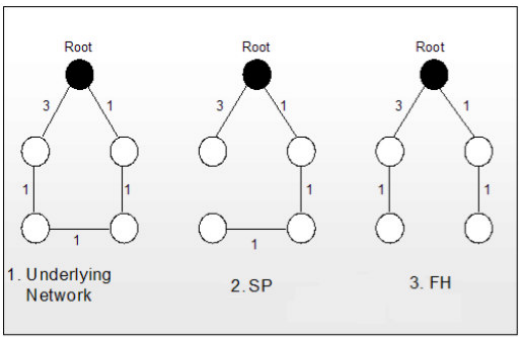


Figure 1: FH and SP Spanning Trees.

SP and FH are just two common spanning tree types and many others are also available. We are currently investigating the performance of a range of spanning trees for off-site data dissemination. Nevertheless, these two examples serve to illustrate the trade-off that often exists between overlay performance and power consumption.

4.4 CPU Power Adaptation

The XScale PXA255 CPUs used in the GridStix platform support software controlled frequency scaling, which allows the CPU to be set at a variety of speeds from 100MHz to 400MHz.

Processing power increases with clock speed but at the cost of increased power consumption. Figure 2 shows the relationship between the clock frequency of the XScale CPU and the power it consumes.

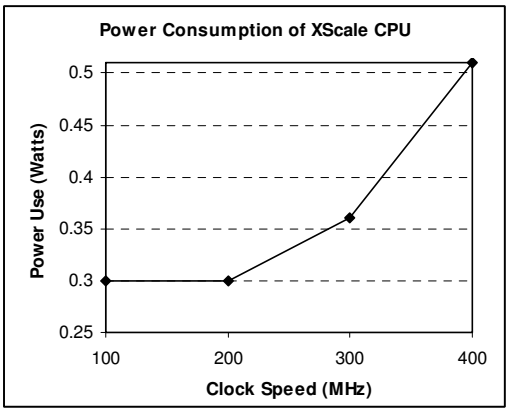


Figure 2: XScale Power Consumption.

Of course, power consumption is also affected by the way the CPU is used by applications and therefore power management could also be implemented by explicitly controlling the manner in which local processes are scheduled (e.g. by modifying the frequency with which local point-prediction models are executed). Nevertheless, the XScale's support for software control of clock frequency provides convenient,

coarse grained adaptation of CPU power consumption.

5. Adaptation Scenarios

5.1 Overview

In the above, we have presented a number of ways in which the behaviour of GridStix nodes can be adapted. However, this is only half of the story—for adaptation to be useful and meaningful it must be suitably informed by relevant real-world triggers.

We now present three adaptation scenarios which demonstrate how awareness of real-world conditions can be used to maintain optimal system operation in changing environmental conditions. The first scenario considers situations in which sensor nodes might become immersed in water; the second considers adapting to total node failure, and the third considers adapting to changes in criticality. These scenarios are not exhaustive; however they do demonstrate how local computation can be used to optimise WSN behaviour and thus produce a more useful and more robust flood prediction system.

5.2 Adapting to Node Immersion

In a flood-prone area, sensor nodes clearly face the risk of immersion. While effort is made to deploy nodes above the likely level of flood water, this is impossible to guarantee. As each GridStix is sealed in a waterproof enclosure, immersion, even in deep water, does not actually damage the node. However, significant immersion prevents power being produced by the solar panels and also adversely affects wireless communication technologies.

Longer wavelength wireless technologies such as GPRS are better at penetrating water than short wavelength technologies such as 802.11b or Bluetooth. Therefore, when a node becomes submerged, or the node predicts that submersion is likely, GPRS communication should be used as this is more likely to result in a sustained connection. During normal operation however, GPRS is the poorest choice for on-site communication due to its low bandwidth, low QoS and high power consumption. Therefore under normal conditions 802.11b or Bluetooth would be used.

Emergent effects of submersion may lead to the need for further adaptations. For example, a lack of power being produced by a node's solar panel might indicate a reduction of CPU clock speed or lowering the frequency with which the flood models are executed.

5.3 Adapting to Node Failure

Alongside the previously-discussed risk of node immersion, sensor nodes are at significant risk of damage or destruction due to being swept away by flood water or due to collision with debris. This risk may be assessed using on-site flow rate measurements. While it is impossible for a failed node to adapt its behaviour in this instance, the impact that a node's failure has on the WSN as a whole is highly dependent on the application level overlay that is being used to disseminate data offsite.

Consider the spanning trees introduced in section 3: Shortest Path (SP) trees consume less power than Fewest Hop (FH) trees and therefore during normal operation, off-site dissemination of sensor data should be performed using an SP spanning tree. However, when on-site flow measurements indicate an increased risk of node failure, the system should switch to an FH spanning tree, which is significantly more resilient to node failure. In this way power-consumption is minimised during normal operation, whilst resilience is preserved at times of high risk.

5.4 Adapting to Changes in Criticality

As previously described, local point predictions are used to provide timely warnings for local stakeholders. When such a warning is in place, the computation of local flood warnings becomes more time-critical. Where initial flood warnings were accurate, local predictions can be used to show likely paths of inundation, and in the case of erroneous warnings, local predictions can be used to lift the flood warning. The latter is particularly critical as flood preparation and particularly evacuation is an expensive activity.

During normal system operation, the timely execution of flood warnings is not particularly critical and therefore, nodes can scale down their CPU speed to 200MHz to minimise CPU power consumption (to 0.3 watts). However, when stakeholder flood warnings are in place and the computation of flood warnings becomes more critical, the nodes can increase their CPU speed to the maximum of 400MHz (0.52 watts). In this way, the system conserves power during normal operation while maintaining the ability to provide timely flood warnings in critical situations.

The rate at which the sensors collect data can also be increased during these critical times. This enables the system to provide more frequent and accurate predictions though at the

cost of increased network, CPU and power resources.

Note that the performance maximisation actions taken when flood prediction becomes time-critical are in direct conflict with the power saving actions taken when battery power runs low (see above). In cases where logical adaptive actions conflict, the system must determine which action is most critical. We are currently in the process of assessing this through evaluation of system performance. Furthermore, this is by no means the only situation in which adaptations interfere with each other. In particular, there are a number of ways in which the type of physical network in use affects the optimal choice of overlay, and vice versa.

6. Deployment and Evaluation

6.1 Deployment

The system described above has already been built and tested in the lab and we are now preparing to deploy it in a real-world environment. The planned deployment site is at Cow Bridge, which is located on the River Ribble in the Yorkshire Dales. This site is prone to flooding for much of the year and thus offers good potential for evaluating the system under real-world conditions. Flooding at the site affects the nearby village of Long Preston, which thus additionally presents us with a motivation for evaluating warning systems for local stakeholders. The site is largely rural which minimises the risk to deployed hardware due to theft and/or vandalism. We anticipate initial deployment during summer 2006, when instances of flooding are relatively uncommon.

Deployment at the site will cover approximately 1km of river with an initial installation of 13 nodes. The majority of sensors deployed will be depth sensors, along with a single image-based flow sensor and a single ultrasound flow sensor.

6.2 Further Testing and Simulation

While the Cow Bridge deployment provides a realistic environment for evaluating system performance, it has a number of significant limitations such as a limited scale of deployment, the unpredictability of flood events, and the time required to perform tests. We are therefore concurrently pursuing lab-based and simulation-based testing of the system to gain more insight into its generic applicability.

In particular, we are currently engaged in assessing the performance of the solar panels in terms of the power they produce in various weather conditions (hours of daylight, cloud cover etc.), and are assessing battery performance in terms of charge retention. In addition, we are assessing the performance of the physical networking hardware (802.11b, Bluetooth and GPRS) in terms of power consumption and QoS characteristics such as throughput, loss, delay and jitter. These factors are being evaluated with various usage profiles in various weather conditions and under varying levels of immersion.

Based upon these basic performance characteristics, a simulator is being constructed that will allow the testing of various application-level networks and power management strategies, using site-specific topographical information, past weather conditions and past flooding data. This will be used to prototype potential deployment technologies and investigate the performance of large-scale deployments that could not be easily tested in the real world.

7. Related Work

A number of grid-related projects have addressed the issues of WSN-grid integration in general and WSN-based flood prediction in particular. A prime example of the former is the Equator remote medical monitoring project [Rodden '05], and a prime example of the latter is Floodnet [DeRoure '06]. However, these systems (to the best of our knowledge) all employ a 'dumb' proxy-based approach to integrating WSNs with the grid and thus cannot take advantage of the local computational power that we employ to drive the adaptation of WSN behaviour, to support richer sensor modalities such as image-based flow prediction, and to provide timely flood warnings to local stakeholders.

8. Conclusions and Future Work

This paper has described a WSN that is capable of performing not only remote off-site flood modelling based on grids in the fixed network, but also local on-site flood modelling using a lightweight grid built on our GridStix platform.

The key difference between our system and existing work on WSN-grid integration is that our work aims to promote the sensors to *first class* grid entities. This allows a greater degree of integration and flexibility than those approaches that treat sensor networks as

conceptually distinct from the grid. In particular, for our flood prediction scenario it allows us to more effectively support WSN adaptation, to support richer sensor modalities, and to enable proactive behaviour such as informing local stakeholders of pending flooding.

In future research we are especially planning to work on improving our system's adaptation mechanisms. Currently, our adaptation policies are manually implemented, but we plan in the future to investigate the extent to which nodes can 'learn' appropriate adaptation behaviour. As an example, consider the performance of power management approaches. If nodes were capable of autonomously selecting appropriate power management strategies, it would significantly reduce the time-to-deployment for novel environmental monitoring applications. To accomplish this, sensor nodes could successively load different power management policies and, based on the relative success of these policies, select the most appropriate one for a given environmental monitoring scenario, or set of environmental conditions.

Currently, the information used to inform system behaviour originates exclusively from within the system itself. However, external information might also provide valuable information on which adaptation could be based. For example, local weather predictions, particularly predicted hours of sunlight, could be used to better inform battery-life models (due to fluctuations in the power captured by solar panels).

A final area of planned future work is to investigate how our WSN's functionality may be expanded from an exclusively monitoring role to additionally encompassing flood-response support. For example, real-time on-site visualisation of flood models would be useful for the emergency services who could use this data to inform the placement of sand bags and other flood defences. Similarly, the digital cameras deployed to perform image-based flow measurement could be switched to providing real-time remote imaging for flood responders. This adaptation of node roles necessitates not only modifications to local functionality, but also imposes new requirements for the supporting physical and application-level networks.

References

[Environment Agency '00] The Environment Agency "Lessons Learned, Autumn 2000 Floods", available online: http://www.environmentagency.gov.uk/commondata/acrobat/126637

[Pappenberger '05] F. Pappenberger,, K. Beven., N. Hunter et al., "Cascading model uncertainty from medium range weather forecasts (10 days) through a rainfall-runoff model to flood inundation predictions within the European Flood Forecasting System (EFFS)", published in Hydrology and Earth System Science, 9(4), pp381-393, 2005.

[Beven '05] Beven, K J, Romanowicz, R, Pappenberger, F, Young, PC, Werner, M, "The Uncertainty Cascade in Flood Forecasting", ACTIF meeting on Flood Risk, Tromsø, in press, 2005.

[Bradley '03] Creutin J. D., M. Muste, A. A. Bradley, S. C. Kim, "River Gauging using PIV Techniques: A Proof of Concept on The Iowa River", A. Kruger, Journal of Hydrology 277, 182-194, 2003.

[Hughes '06] Hughes D., Greenwood P., Coulson G., Blair G., Pappenberger F., Smith P., Beven K., "GridStix: Supporting Flood Prediction using Embedded Hardware and Next Generation Grid Middleware", to be published in the proceedings of the 4th International Workshop on Mobile Distributed Computing (MDC'06), Niagara Falls, USA, June 2006.

[Coulson '05] G. Coulson, P. Grace, G. Blair et al, "Open Overlay Support for the Divergent Grid", in the proceedings of the UK E-Science All Hands Meeting, Nottingham, UK, September 2005.

[Coulson '02] G. Coulson, G. Blair, M. Clark et al "The Design of a Highly Configurable and Reconfigurable Middleware Platform", in the ACM Distributed Computing Journal, Vol. 15, No 2, pp109- 126, April 2002.

[Rowstron '01] A. Rowstron, P. Druschel. "Pastry: Scalable, Decentralised Object Location and Routing for Large Scale Peer-to-Peer Systems" – Conference on Distributed Systems Platforms, Heidelberg, Germany 2001.

[Gnutella '00] Limewire, "The Gnutella Protocol Specification v0.4", available online: http://www9.limewire.com/developer/gnutella_protocol 0.4.pdf

[XBow '06] Crossbow "Mica Mote Data Sheet", available online: www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA.pdf

[Waysmall '06] Waysmall Computers "Gumstix Embedded Computing Platform Specifications", website: http://gumstix.com/spexboards.html.

[Cooper '05] "The Open Overlays Collaborative Workspace Environment", Chris Cooper, David Duce, Muhammad Younas, Wei Li, Musbah Sagar, Gordon Blair, Geoff Coulson, Paul Grace, ", In Proceedings of the UK E-Science All Hands Meeting 2005, Nottingham, September 2005.

[Conti '05] "A Cross-layer Optimization of Gnutella for Mobile Ad hoc Networks", Conti M., Gregori E., Turi G., Published in the proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '05), IL, USA, 2005.

[DeRoure '05] D. DeRoure, **"**Improving Flood Warning times using Pervasive and Grid Computing", submitted to quarterly of Royal Academy of Engineering, UK.

[Rodden '05] T. Rodden, C. Greenhalgh, D. DeRoure, A. Friday, "Extending GT to Support Remote Medical Monitoring," in the proceedings of UK e-Science All Hands Meeting, Nottingham, UK, September 2005.

[Ericsson '02] "Ericsson ROK 104 001 Data Sheet", available online: http://prism2.mem.drexel.edu/~billgreen/Bluetooth/rok104001.pdf

[Ambicom '06] "GPRS Compact Flash Card For Laptop And PDA Specifications" available online: http://www.ambicom.com/products/gprs/gprs-feat.html

[Dijkstra '59] E. W. Dijkstra: "A note on two problems in connection with graphs" in: Numerische Mathematik. 1 (1959), S. 269–271

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Supporting the Clinical Trial Recruitment Process through the Grid

Anthony Stell, Richard Sinnott, Oluwafemi Ajayi
National e-Science Centre
University of Glasgow, UK
ajstell@dcs.gla.ac.uk

Abstract

Patient recruitment for clinical trials and studies is a large-scale task. To test a given drug for example, it is desirable that as large a pool of suitable candidates is used as possible to support reliable assessment of often moderate effects of the drugs. To make such a recruitment campaign successful, it is necessary to efficiently target the petitioning of these potential subjects. Because of the necessarily large numbers involved in such campaigns, this is a problem that naturally lends itself to the paradigm of Grid technology. However the accumulation and linkage of data sets across clinical domain boundaries poses challenges due to the sensitivity of the data involved that are atypical of other Grid domains. This includes handling the privacy and integrity of data, and importantly the process by which data can be collected and used, and ensuring for example that patient involvement and consent is dealt with appropriately throughout the clinical trials process. This paper describes a Grid infrastructure developed as part of the MRC funded VOTES project (Virtual Organisations for Trials and Epidemiological Studies) at the National e-Science Centre in Glasgow that supports these processes and the different security requirements specific to this domain.

1. Introduction

To test new drugs and treatments for clinical care requires careful and long-term testing before they can be prescribed to the population in general. To facilitate such testing requires identification and recruitment of large groups of the population that fit certain criteria related to the specific condition that the drug is addressing.

Automating this process has numerous advantages including reduced cost and expediting the clinical trials process as whole, by avoiding unnecessary contacts with non-suitable members of the public. An example of this is where the recruitment criteria require candidates with a cholesterol level between certain bands. Such information is not typically known by the vast majority of the public. Weeding out patients outside of the needed bands is therefore beneficial. Over-recruiting is also advantageous since it may well be the case that potential candidates may decline to be involved in a given trial, or alternatively that certain candidates are better matches than others.

In order to co-ordinate such a trial process requires the combined effort of numerous personnel with specific roles in the whole process: clinical trials investigators and nurses wishing to recruit patients for a given trial; ethical oversight committee members and Caldicott guardians responsible for deciding on the ethical aspects of the study; clinicians and general practitioners (GPs) responsible for individual patients; and importantly the patients themselves. Before any access to identifying patient data sets is made, it is necessary to obtain patient consent for the use of a given patients' personal and private medical data. The interplay between these roles and the process by which a clinical trial is co-ordinated is crucial to the overall success, viability and legality of a trial.

Given the fact that the clinical data sets are typically scattered across many resources and institutions, including GP databases, hospitals, and disease registries amongst others, Grid technology, in principle, provides many potential advantages to deal with data federation. However, this domain also has numerous challenges, especially related to security, which must be explicitly addressed. Due to the sensitive and confidential nature of the data in this domain, strict controls are required on access and distribution, with only sufficiently privileged actors having the appropriate levels of access.

The VOTES project (Virtual Organisations for Trials and Epidemiological Studies) [1] has been funded by the Medical Research Council (MRC) to explore this problem space. One of the focuses of VOTES, and the primary focus of this paper, is to develop Grid solutions that address large-scale recruitment needs in the clinical domain. In addition VOTES is addressing two other important areas in the support of clinical trials and epidemiological

studies: data collection and study management. Furthermore, it is a requirement that the infrastructure that will be developed will be effective yet simple to use for the non-Grid personnel involved in the clinical trials process.

2. Clinical Patient Recruitment

As described previously, clinical patient recruitment is a large-scale and resource-consuming exercise. The human challenge in co-ordinating such a large effort can be immense and in some cases, such as the UK Biobank Project [2] the number of potential candidates is so large that the use of distributed technology is mandatory for the task to be completed in a meaningful time-scale. (UK Biobank expects to recruit 500,000 members of the population between 40-69 years of age).

2.1 Crossing Domain Boundaries

To effectively identify suitable trial candidates on a large a scale requires knowing the structure of patient information data sets across a broad set of domains. For instance, to sample the national population of the UK would require knowing the data structures of the health services in England and Scotland, knowing how they relate to each other and knowing how to translate between the schema of both.

Ideally there should be a single electronic health record which captures all necessary health information associated with a given patient that is accessed and updated by all health care providers throughout a patients' lifetime. This should support tracking of a patients' place of residence throughout their lifetime, and allow for cross checking of records in one area to those of the other. However such a single e-Health record remains a distant wish and a variety of heterogeneous and largely non-interoperable legacy infrastructures and data sets is the norm across the NHS, with paper based patient case histories and records still commonly used.

To support the linkage of distributed data sets associated with a given patient, it is beneficial to have a common, unique identifier for patients that spans all domains and can be used to join the patient records between databases. In Scotland this unique identifier is a number known as the Community Health Index (CHI), which is currently being rolled out across the nation as part of a new Scottish parliamentary initiative. It is planned that the CHI number will be rolled out across all of Scotland by mid-2006. In England, the unique identifier is the NHS Number, which is a distinct entity from the CHI. Relating these two numbers, which have different structures in different contexts, is a major, yet unavoidable, challenge.

2.2 Recruitment Work-Flow

A patient recruitment process must ideally capture patient consent as early as possible. Whilst information on patients is stored in a variety of digital formats and locations, *a priori* consent that these data sets can be accessed and queried to decide that a given patient be recruited to a clinical trial, is needed. One of the best sources of information associated with potential trials candidates is through primary care sources, i.e. in their GPs databases. Understanding whether a given clinical trial is in the interest of a particular patient is best answered by these GPs.

Figure 1 gives a pictorial representation of the process/workflow through which a clinical trial investigator and a GP might interact to support primary care recruitment.

1. The trial co-ordinator wishes to set up a new clinical trial, with a specific description of the drug/treatment and the patients' characteristics that would potentially qualify them for entry into the clinical trial. They need to contact a set of GPs to describe this new trial and find out if these GPs have patients that fit the required criteria.
2. Assuming that the GP is interested in participating in this particular trial, they need to search their own patient records for anyone that may fit these criteria.
3. Assuming that one or more matching patients have been found, the GP decides if it is really in the patients' interest to participate in the clinical trial. If this is the case then said patient is contacted by the GP and told about the trial. If they are willing to participate having had the potential benefits and issues that might be associated with the trial fully explained, they are asked for their consent to use their personal data for this purpose. Once consent is obtained, this information is recorded.
4. Based upon the specifics of the clinical trial, various data sets are collected from the GPs database. These can be non-identifying information such as age, height, weight, medical conditions and social and demographic details. Identifying information may be anonymised with the de-anonymising keys maintained by the GPs.
5. This information along with the note of consent is communicated back to the trial

co-ordinator. The information is then validated and stored for later use in the trial.

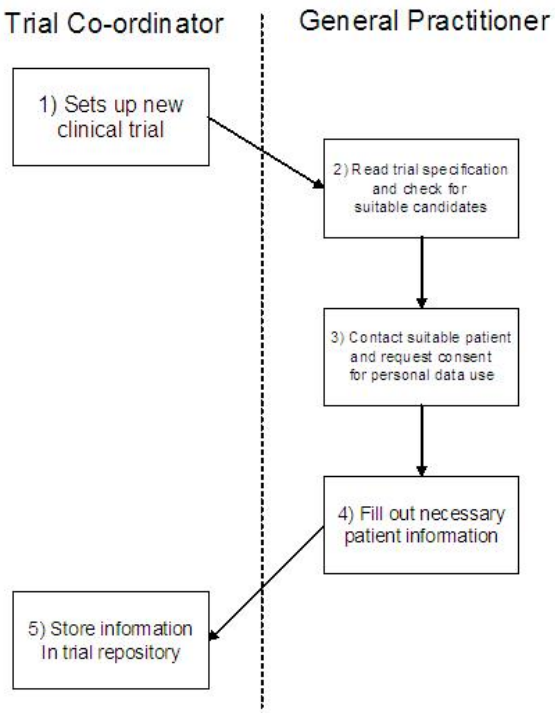


Figure 1: A diagram of the primary care patient recruitment workflow.

One of the central requirements of this process is the need to clearly separate the distinct duties of the two actors involved. There is necessarily a disparity between the privileges of the two roles, and the data that they will respectively be allowed to access, and at which particular times. To enforce this type of interdependent and role-based access control requires a sophisticated security system.

3. Grid Security for Clinical Trials

Grid computing depends on the collaboration and sharing of resources across domain boundaries. A loose coupling of resources and user access to achieve a specific goal over a set period using Grid technology is often termed a Virtual Organisation (VO).

Traditionally, Grid security has been expressed in terms of access control between nodes within a VO. In this VO, sites have only a limited amount of trust between each other, yet they also wish to share certain ring-fenced resources that will allow the VO to accomplish its overall goal.

However, enforcing data security in the clinical and health-care domain is a more complex problem. While the tenets of Grid security apply – that of Authentication, Authorization and Accounting ("AAA") – there are other more subtle requirements that must also be met when attempting to set up a system that is flexible enough to use Grid technologies effectively, but also maintains the high standards of privacy and integrity required in the clinical domain. A term used to describe this particular entity is a Clinical Virtual Organisation (CVO). The level of prescription of security policies and how they are enforced must be strongly adhered to within such a CVO.

3.1 Clinical Security Considerations

Clinical security can be broadly divided into the following three areas [3]:

- Sensitivity
- Consent
- HCP (HealthCare Professional) speciality

Sensitivity relates to the importance and privacy level applied to a data field within a health-care record. Considering risk analysis, it can be described in terms of the possible consequences if privacy of this data record is broken. The level of sensitivity will ultimately be determined by the HCP dealing with the particular record, which in turn relates to the speciality of that HCP (see below). Where the records are being used for statistical aggregation of data, a corollary of sensitivity is the need for "anonymisation" of that data – where any information that identifies a patient must be hidden from unprivileged users.

Consent relates to the requirement of asking a patient whether they will allow their information to be used in this clinical trial. There is a subtlety in obtaining consent as to what parts of the patient's record they will release for use and which they wish to remain private. To allow this "pick and mix" release of consent gives a more flexible structure. However, the patient must be guided in this by the GP, to provide a professional opinion on the consequences of releasing this consent on differing parts of their data records, with the possibility therein of differing levels of sensitivity.

HCP Speciality refers to the many different categories of HCP that can exist. This affects access to data records as one HCP may have rights to see highly sensitive records in their own fields but not in another. This speciality will be classified within the security policy that defines privileges within the VO.

Another consideration that is idiosyncratic to security within health-care is the scenario sometimes called the "broken glass" situation. This is where highly sensitive data is accessed by an HCP that does not have the necessary privileges, in an attempt to save a patient's life. In the immediate situation, the HCP believes it

necessary to access this record, and time is of the essence. The system here is to let the HCP have access to this information but have an irreversible record that this unprivileged access has occurred. When the immediate situation is resolved the logs of the event should be investigated by an auditing authority, to see whether the actions of the HCP were justified.

3.2 Clinical Data Security Policies

A security policy defined in one site node of a VO will not necessarily have the same structure as a different node within the same VO. This is inherently tied to the structure of object classification within a specific domain, as security can only be defined and enforced on data that itself has a well-defined structure. Put another way, sites must have their own autonomy and hence define and enforce their own security policies on access to different data sets.

There are numerous developments in standards for the description of data sets used in the clinical domain however. These are complex and evolving with numerous commercial bodies and standards groups involved in developing strategies and include major initiatives such as Health-Level 7 (HL7) [4], SNOMED-CT [5] and OpenEHR [6]. There is often a wide range of legacy data sets and naming conventions which impact upon standardisation processes and their acceptance. The International Statistical Classification of Disease and Related Health Problems version 10 (ICD-10) is used for the recording of diseases and health related problems and is supported by the World Health Organisation. In Scotland ICD-10 is used within the NHS along with ICD version 9. ICD-10 was introduced in 1993, but the ICD classifications themselves have evolved since the 17th Century [7].

To compound this problem of classification, there is the issue of the dynamic nature of virtual organisations. One of the standard characteristics of a VO is that it is not only a loose collaboration between sites but it is also a transient one, with a limited lifetime and for a specific purpose. As such, any security policy enforced will also have a limited lifetime – requiring the re-evaluation of security requests after a given time period. This must be considered when defining security policies and establishing chains of trust.

3.3 Grid Security Solutions

Security solutions in the Grid community are largely categorised by where they fit into the "AAA" scenario.

Authentication – this is almost always achieved using Public Key Infrastructures (PKIs) where public and private keys and certificates are used to verify the authenticity of a user's identity.

Authorization – as this is a more complicated requirement, in terms of establishing privilege rights based on identity, there is a wider range of possible solutions in the community (PERMIS [8], CAS [9], VOMS [10], Akenti [11]). These applications all have various advantages and disadvantages depending on the needs of the developer and the implementation idiosyncrasies, but no clear leader has yet been established in the field. In the VOTES project, authorization is provided by a simple implementation of an Access Control Matrix (see section 4.3).

Auditing – this is a security measure that has more relevance later in the production cycle of a system. Whilst not underestimating the importance of logging all user activity and being able to attach events to individuals, design and production in the VOTES project is currently focused on securing access to the system in the first instance and supporting the recruitment process.

4. VOTES Implementation

The details of the portal application that is currently under development to provide solutions to these security, usability and process challenges, are now described in this section.

4.1 Grid Technologies

The implementation of the VOTES project has built largely upon the expertise in certain Grid technologies based at the National e-Science Centre in Glasgow. These include the following applications that provide tools to develop and maintain the application: GridSphere, Globus and OGSA-DAI.

GridSphere [12] is a web portal technology that provides easy access to secure grid services. It can be presented as a group of layered portlets allowing simultaneous, and interactive, task processing. A development suite is available that provides easy-to-use tools and tutorials for building and deploying these portal services.

The Globus Toolkit [13] similarly provides a set of distinct modular tools that allow development of Grid services. Version 4.0 of the toolkit is implemented in the VOTES project, which has been developed to the Web Services Resource Framework (WS-RF) specification [14] - this is in line with a drive by Globus to align their service architecture to the more common web services standard [15].

OGSA-DAI [16] (Open Grid Services Architecture – Data Access and Integration) is a project that has developed a toolkit specifically for grid services that federate data from distributed sources. The OGSA-DAI services developed in VOTES again follow the WS-RF specification, but this is implemented in addition to direct JDBC connections to local data sources. The major advantage of using OGSA-DAI is that many different data sources, from DBM systems to XML and flat files, can be queried. The data returned is then rendered in a standard XML format, and tools are provided to allow easy analysis and presentation of this format.

4.2 GPASS and SCI Store

In the VOTES implementation thus far, the development team have been given access to relevant software and realistic (representative) data sets used to explore the functionality of the NHS services and data. Negotiations are on-going in rolling out the services described in the following sections across the wider NHS in Scotland and in gaining access to actual clinical data sets

The General Practice Administration System for Scotland (GPASS) [21] is a client-side application that is in use by a large number of general practitioners, in over 890 practices in Scotland. It provides a portable, 'on-the-spot' electronic interface for GPs to input patient data, to be synchronised to the centralised SCI-Store [22] repository at such times as a connection can be made. It also contains a local data store that houses and inter-relates information on drugs and treatments, thus providing a rudimentary ability to advise on a particular treatment, depending on patient symptoms and history [23]. The interface used to show the electronic record of patient history is shown in Figure 2.

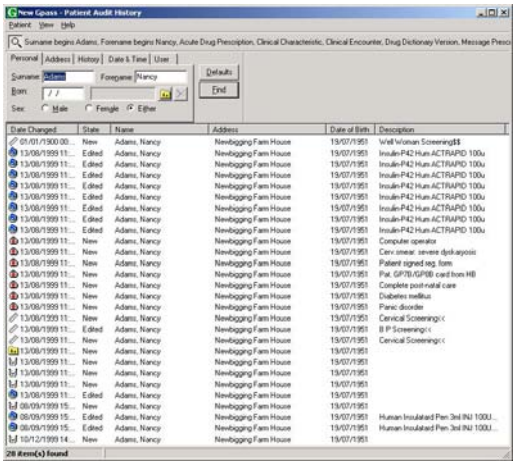


Figure 2: A screenshot of the GPASS patient history function.

Because of its proximity to the 'front-line' of primary care, GPASS is the most pertinent technology when looking at patient recruitment for clinical trials. The distributed and asynchronous structure of the application makes the process of integrating this securely into a wider system a challenging but necessary one.

The Scottish Care Information (SCI) Store [22] is a batch storage system which allows hospitals to add a variety of information to be shared across the community, e.g. pathology, radiology, biochemistry lab results are just some of the data that are supported by SCI Store. Regular updates to SCI Store are provided by the commercial supplier using a web services interface. Currently there are 15 different SCI Stores across Scotland (with 3 across the Strathclyde region alone). Each of these SCI Store versions has their own data models (and schemas) based upon the regional hospital systems they are supporting. The schemas and software itself are still undergoing development.

SCI Store serves as a repository for this data across regional and national domains. As individual practitioners update their GPASS databases, the information is automatically uploaded to this central repository. The VOTES infrastructure makes use of this by developing grid services that allow interrogation of the centralised SCI Store repository when collecting study data, and also allow interrogation of the individual practitioner databases for patient recruitment. An interface between the grid services already developed in VOTES and a test GPASS database is shown in Figure 3.

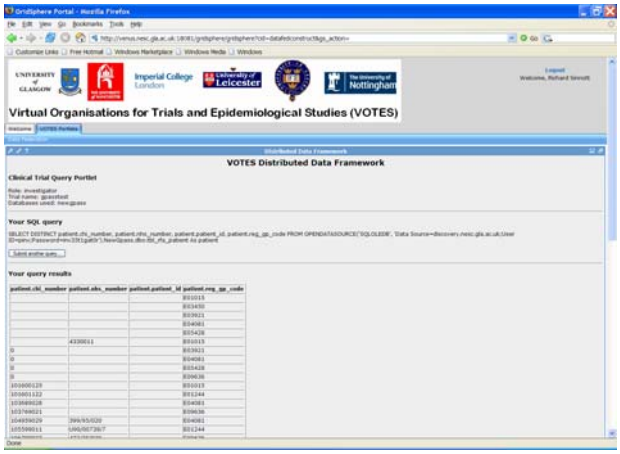


Figure 3: Results returned from a query to a local installation of GPASS.

The security implications in being able to query the individual databases used by GPASS are discussed in section 4.4.

4.3 VOTES Portal

The central theme of the VOTES project is to set up a Clinical Virtual Organisation (CVO) that will implement the three-fold vision of patient recruitment, data collection and study management.

In the context of primary care patient recruitment, the VOTES portal provides web access based on the role of either "Trial Co-ordinator" or "General Practitioner". (For data collection and study management a wider range if roles is supported). The grid and data services behind the portal provide distributed methods of:

- retrieving the necessary trial, patient and GP information.
- retrieving and storing the trial forms
- allowing asynchronous communication between the co-ordinator and the GP in the work-flow (see section 2.2).

An outline of how these scenarios are supported is depicted in the VOTES portal infrastructure shown in Figure 5.

The envisaged future development of the VOTES portal includes the addition of repeated modules in the overall architecture, including extra portal and data servers. (To see an outline of the current architecture, see [1].) This will allow a more "Grid-like" structure to be developed, providing features necessary in any production system, such as redundant failover, and also features specific to Grid technology such as intelligent load-balancing and distributed server functionality based on the resources available at specific nodes.

4.4 VOTES Security

NeSC-Glasgow has extensive experience in a range of fine-grained authorisation infrastructures across a range of application domains [17-19]. Whilst it is expected that the existing prototype will be moved to a more robust authorisation solution, the following authorization infrastructure has been developed, based on an access matrix as shown in Figure 4. This allows for rapid prototyping, which allows the problem space to be explored and user feedback to be obtained as early as possible.

		R_1	R_2	R_3	R_4
		h_1	h_2	h_3	h_1
U_1	U_1	0	0	1	0
U_2	U_2	0	0	0	1
U_3	U_3	1	1	1	1
U_4	U_4	0	1	0	0

$U_1(R_1 \Delta h_3) = 1$, $U_2(R_1 \Delta h_2) = 0$, $U_3(R_3 \Delta h_1) = 1$, $U_4(R_2 \Delta R_3 \Delta h_4) = 0$,
where Δ is a combination function, 0, 1 are bit-wise privileges, R_X, h_X are resources and U_x is a subject

Figure 4: Access Matrix Model

The authorisation mechanism implements an Access Matrix model [20] that specifies bit-wise privileges of users and their associations to data objects within the CVO. Data objects are defined as fields, tables, views, databases and sites, for the purposes of fine-grained authorisation. The access matrix is designed to enforce discretionary and role based access control policies. It is also scaleable to facilitate ease of growth parallel to the predicted growth of the infrastructure as a whole.

The NeSC at Glasgow have already shown in numerous other works [27,28] how Grid services can be protected through technologies such as GSI and PERMIS, however the effort in supporting these infrastructures is considerable and not conducive to rapid prototyping necessary to capture the *basic* functionality needed in clinical trials. Once the access matrix model has allowed for the detailed expression and enforcement of policy which the clinicians and all people involved in the clinical trials process are satisfied with, a move to a full RBAC model may well be considered depending upon the strategic direction of the project.

Security on the data sources is achieved at both local and remote level. The local level security, managed by each test site, filters and validates requests based on local policies at the DBMS level. The remote level security is achieved by the exchange of access tokens between the designated Source of Authority (SOA) of each site. These access tokens are used to establish remote database connections between the sites in the federation. In principle local sites authorise their users based on delegated remote policies. This is along the lines of the CAS model [9].

Considering security in GPASS, it is probable that due to the distributed nature of the application, a modification to the security model adopted so far in VOTES will need to be made in future development. The technology used in VOTES so far, in particular the portal's ability to query and return results from the back-end GPASS database, shows that it is possible to implement a web/grid service interface that provides a handle for third parties to securely interrogate GPASS.

However, until significant progress is made between the participating agencies in VOTES, it is unlikely that this interface will be adopted by the users and developers of GPASS. This is part of the human and political factor that must be overcome before the technology in VOTES will be taken up. In particular it is intended that the prototypes will be explored with sets of GP practices across the Greater Glasgow region through the SPPIRe network [26].

5. Conclusion

The prototype application developed in the VOTES project is currently a work in progress. It does not yet provide all the answers to the issues posed in this paper, but it does provide a starting place, and is being designed with the larger scheme in mind.

Using the current implementation it is possible to envisage a system that will identify potential trial candidates quickly, securely and efficiently. It is to be hoped that with the use of such electronic methods, the scope for error, such as mis-identification of patients or release of confidential information, will be very much reduced. A key aspect of the work is to support the capture of patient consent as early as possible in the clinical trial recruitment process. Scenarios where statistical information from GPASS is retrieved where, once consent is given through a combination of patient and GP interactions, more detailed information is returned are under development.

As with most research that is addressed using Grid Computing, only some of the cross-domain issues are to do with the technology. A lot depends on the human and political factors between participating bodies. These issues take time and the establishment of trust to be overcome. However, it is hoped that systems such as the one described in this paper will allow a closer and more "joined-up" network of clinicians and technologists to promote that trust and encourage closer collaborative work. To support this, it is planned that the researchers working on the VOTES project will also be given honorary contracts to work part time in the NHS in Glasgow.

6. References

[1] Virtual Organisations for Trials and Epidemiological Studies (VOTES) – http://www.nesc.ac.uk/hub/projects/votes
[2] UK Biobank project – http://www.biobank.ac.uk
[3] Roger Quin, NHS Greater Glasgow – Personal Communication
[4] Health Level 7 (HL7) – http://www.hl7.org
[5] SNOMED-CT – http://www.snomed.org/snomedct
[6] OpenEHR – http://www.openehr.org
[7] ICD background, http://www.connectingforhealth.nhs.uk/clinicalcoding/faqs
[8] PERMIS – http://sec.isi.salford.ac.uk/permis
[9] CAS – http://www.globus.org/toolkit/docs/4.0/security/cas
[10] VOMS – http://hep-project-Grid-scg.web.cern.ch/hep-project-Grid-scg/voms.html
[11] Akenti – http://dsd.lbl.gov/Akenti
[12] GridSphere – http://www.Gridsphere.org
[13] Globus – http://www.globus.org
[14] Web Services Resource Framework – http://www.globus.org/wsrf
[15] WS-RF specification v1.2 - http://www.oasis-open.org/specs/index.php#wsrfv1.2
[16] OGSA-DAI – http://www.ogsadai.org.uk
[17] R.O. Sinnott, M. M. Bayer, J. Koetsier, A. J. Stell, Grid Infrastructures for Secure Access to and Use of Bioinformatics Data: Experiences from the BRIDGES project, submitted to the 1st International Workshop on Bioinformatics and Security (BIOS '06), Vienna, April, 2006
[18] R.O. Sinnott, M. Bayer, D. Berry, M. Atkinson, M. Ferrier, D. Gilbert, E. Hunt, N. Hanlon, Grid Services Supporting the Usage of Secure Federated, Distributed Biomedical Data, Proceedings of UK e-Science All Hands Meeting, September 2004, Nottingham, England
[19] R.O. Sinnott, A. J. Stell, J. Watt, Experiences in Teaching Grid Computing to Advanced Level Students, Proceedings of CLAG+Grid Edu Conference, May 2005, Cardiff, Wales
[20] R. S. Sandhu and P. Samarati, "Access control: principles and practice" IEEE Communications Magazine, vol. 32, no. 9, pp. 40-48, 1994.
[21] GPASS – http://www.show.scot.nhs.uk/gpass
[22] SCI Store – http://www.show.scot.nhs.uk/sci/products/store/SCI Store_Product_Description.htm
[23] Paul Woolman, NHS Scotland Information Services – Personal Communication
[24] SMR – http://www.show.scot.nhs.uk/indicators/SMR/Main.html
[25] NHS Data Dictionary – http://www.isdscotland.org
[26] Scottish Practices and Professionals Involved in Research (SPPIRe) network, http://www.nes.scot.nhs.uk/SSPC/SPPIRe/
[27] R.O. Sinnott, A.J. Stell, D.W. Chadwick, O.Otenko, Experiences of Applying Advanced Grid Authorisation Infrastructures, Proceedings of European Grid Conference (EGC), LNCS 3470, pages 265-275, June 2005, Amsterdam, Holland.
[28] A.J. Stell, R.O. Sinnott, J. Watt, Comparison of Advanced Authorisation Infrastructures for Grid Computing, Proceedings of International Conference on High Performance Computing Systems and Applications, May 2005, Guelph, Canada.

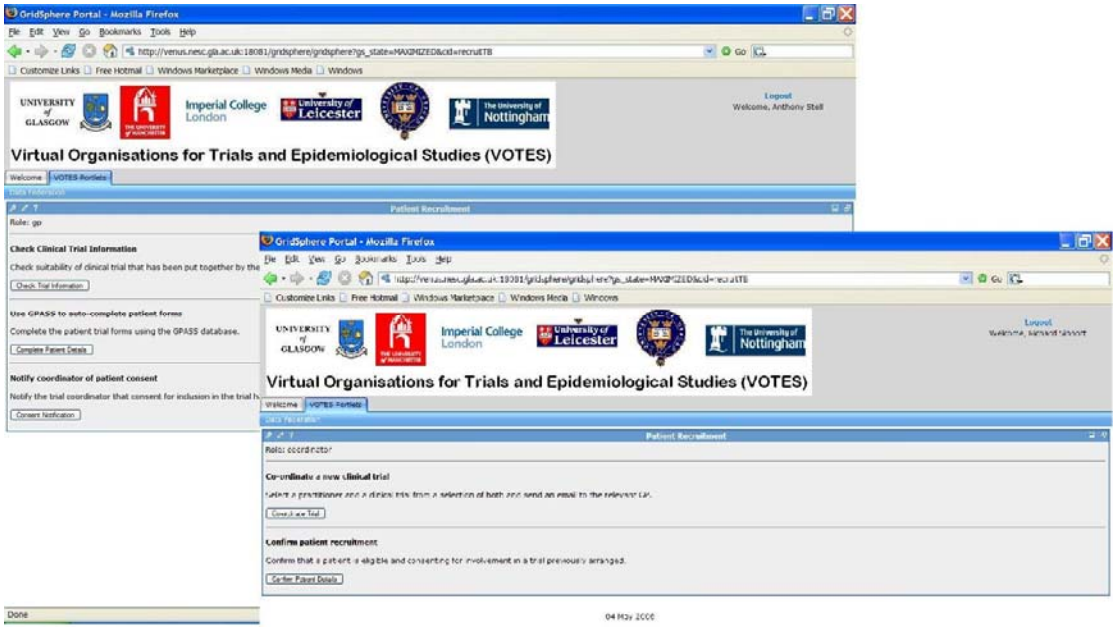


Figure 5: Two views of the recruitment portal. The left image shows the options available in the portal if the role is that of the GP. These options are to check the information on the clinical trial, to use GPASS to auto-complete the patient information and to notify the trial coordinator that patient information and consent has been obtained. The right image shows the coordinator role, which has two options, one to initiate the organisation of the trial and one to upload the final data for the patient.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Summarisation and Visualisation of e-Health Data Repositories

Catalina Hallett, Richard Power, Donia Scott
Computing Research Centre
The Open University
{C.Hallett, R.Power, D.Scott}@open.ac.uk

Abstract

At the centre of the Clinical e-Science Framework (CLEF) project is a repository of well organised, detailed clinical histories, encoded as data that will be available for use in clinical care and in-silico medical experiments. We describe a system that we have developed as part of the CLEF project, to perform the task of generating a diverse range of textual and graphical summaries of a patient's clinical history from a data-encoded model, a *chronicle*, representing the record of the patient's medical history. Although the focus of our current work is on cancer patients, the approach we describe is generalisable to a wide range of medical areas.

1 Introduction

Records of cancer patients are very rich: in addition to a thousand or more numeric data points arising from successive laboratory tests and a chronology of five or six hundred significant events – such as the dates tests were requested or performed, clinics attended or drugs dispensed – our typical patient files will also contain between fifty and a hundred and fifty narrative clinic letters, together with a similar number of reports interpreting a variety of investigations (e.g., Xray, body scan, etc.).

The computer readable part of an electronic patient record for direct clinical care is, therefore, a record of multiple events with no explicit semantic links between them: it records most of what was done, but very little of why. As a result, much if not most of the valuable clinical information remains machine unreadable, locked within the narrative letters and reports exchanged between doctors.

One of the aims of the Clinical e-Science Framework (CLEF) project (Rector et al., 2003), under which the research reported here is being conducted, is to establish a technical infrastructure for managing research repositories of aggregated patient data arising from routine medical care across potential multiple sites and institutions, in support of biomedical research. Information is extracted from medical narratives[1] and aggregated with structured data in order to build complex images of a patient's medical history which model the story of how patient illnesses and treatments unfolded through time: *what* happened, *when*, *what* was done, *when* it was done and *why*. The resulting complex semantic network, termed by us a *chronicle*, allows the construction of targeted summarized reports which do more than present individual events in a medical history: they present, in coherent text, events that are semantically and temporally linked to each other.

This paper discusses the problem of presenting aggregated clinical data: assuming the full richness of clinical information could be made available – whether extracted from clinical records in their current form or acquired a priori using an entirely different data capture paradigm (e.g., structured data entry) – how might that information be represented and exploited for the maximal benefit of clinical research and clinical care? Of particular interest to us here is the problem of automatically generating targetted and comprehensible textual reports from the data-encoded view of a patient's medical history.

In presenting medical histories we are trying to circumvent the shortcomings of textual reports

[1] Using Natural Language Processing techniques, see (Harkema et al., 2005).

by combining them with visual navigation tools. In this way, we take advantage of the better accesibility and interactivity offered by visual timelines as well as of the ability of natural language to convey complex temporal information and to aggregate numerical data.

2 Types of report

The intended end-user of the generated reports is a GP or clinician who uses electronic patient records at the point of care to familiarise themselves with a patient's medical history and current situation. A number of specific requirements arise from this particular setting:

- Events that deviate from the norm are more important than normal events (e.g., an examination of the lymphnodes that reveals lymphadenopathy is more important than an examination that doesn't). However, normal events should also be available on demand.
- Some events are more important than others and they should not only be included in the summary but also highlighted (through linguistic means, colour coding, graphical timelines or similar display features).
- Having different views of the same data is a useful feature, because it allows the clinician to spot correlation between events that they may have missed otherwise.
- Summaries that provide a 30-second overview of the patient's history are often desireable; ideally, these should fit entirely on a computer screen. However, users should be able to obtain more detailed information about specific events by expanding their description.

Following these requirements, we proposed in this project an integrated visualisation tool where users can use a graphical interface coupled with a text generation engine in order to navigate through patient records. Textual reports have the advantage of offering a snaphsot view of a patient's history at any point in time, they can be used for checking the consistency of a patient's record, can be ammended and printed, used in communication between clinicians or clinicians and patients. Text is a good way of describing temporal information (events that happened at a certain position in time with respect to another event), of summarising numerical data (for example, specifying that *liver tests were normal* instead of listing individual measurements for billirubin concentration, Alanine aminotransferase, Alkaline phosphatase, Aspartate aminotransferase, albumin and total protein). However, pure text is not always the best medium of presenting large amount of information, part of which is numerical and most of which is highly interconnected. Text loses the ability of navigating through information, of expanding some events and of highlighting important dependencies between pieces of information. A textual report alone cannot effectively combine the time sequence element with the semantic dependencies - both of which are essential in representing patient records. Depending on the type of report chosen, either one or the other of these elements will necessarily be emphasised at the expense of the other.

We envisage therefore that, depending on circumstances, users may want to have fully textual reports (for example, for producing printed summaries of a patient's history) or combined graphical and textual reports (for interactive visualisation). In the following, we will describe the two reports generated in either of the two scenarios. Section 3 will describe in more detail the natural language generation techniques employed in generating both independent textual reports and report snippets that support the graphical interface.

2.1 Textual reports

Textual reports are views of a data-encoded electronic patient record (a chronicle), which is itself both a distillation and an integration of the elements within the traditional EPR. In this respect, they do not correspond to the narratives traditionally contained in a patient record, such as letters from clinicians, discharge notes, consult summaries. They are a new type of text that aggregates information from the full record.

Based on our requirements analysis with clinicians, we identified two main types of textual report that could be used in different settings. The first is a longitudinal report, which is meant to provide a quick historical overview of the patient's illness, whilst preserving the main events, such as diagnoses, investigations and interventions. It describes the events in the patient's history ordered chronologically and grouped according to the type. It contains most events in the history, although some preliminary filtering is performed to remove usually a small number of isolated events. The following example displays a fragment of a generated longitudinal summary.

```
(1)  The patient is diagnosed with grade 9
invasive medullary carcinoma of the breast.
She was 39 years old when the first malignant
cell was recorded.  The history covers 1517
```

weeks, from week 180 to week 1697. During this time, the patient attended 38 consults.

YEAR 3:

Week 183

- Radical mastectomy on the breast was performed to treat primary cancer of the left breast.
- Histopathology revealed primary cancer of the left breast.

Week 191

- Examination revealed no enlargement of the liver or of the spleen, no lymphadenopathy of the left axillary lymphnodes,no abnormality of the haemoglobin concentration or of the leucocyte count.
- Radiotherapy was initiated to treat primary cancer of the left breast.
- ...

The second class of summary focuses on a given type of event in a patient's history, such as the history of diagnoses, interventions, investigations or drug prescription. In contrast to the longitudinal summaries, which are generic, this type of report is query-oriented, since it summarizes only events which the user deems relevant.

A summary of the diagnoses, for example, will focus on the `Problem` events that are recorded in the chronicle, whilst other events only appear if they are directly related to a `Problem`. This type of summary is necessarily more concise, since the events do not have to appear chronologically and thus can be grouped in larger clusters. Secondary events are also more highly aggregated. For example:

(2) In week 483, histopathology revealed primary cancer of the right breast. Radical mastectomy on the breast was performed to treat the cancer.
In week 491, no abnormality of the leucocyte count or of the haemoglobin concentration, no lymphadenopathy of the right axillary lymphnodes, no enlargement of the spleen or of the liver and no recurrent cancer of the right breast were revealed. Radiotherapy was initiated to treat primary cancer of the right breast.
In the weeks 492 to 496, five radiotherapy cycles were performed.

A subclass of reports in this category is represented by reports of selective events. For example, a clinician may suspect that a certain patient has interrupted their chemotherapy package repeatedly and wants to see if this correlates with a certain medical condition such as anaemia or if there are other causes behind it. In this case, they may order a report focused on incomplete chemotherapy packages and investigations of type blood test.

2.2 Visual reports

A visual report is a one-screen overview of a patient record (see fig.1), where various types of events are colour-coded and displayed along a timeline. Selection of events can be used for highlighting event dependencies or for generating focused textual reports. Apart from general history timelines, users can also investigate the trend of numerical values, for example increases and decreases in the billirubin concentration from one test to an other.

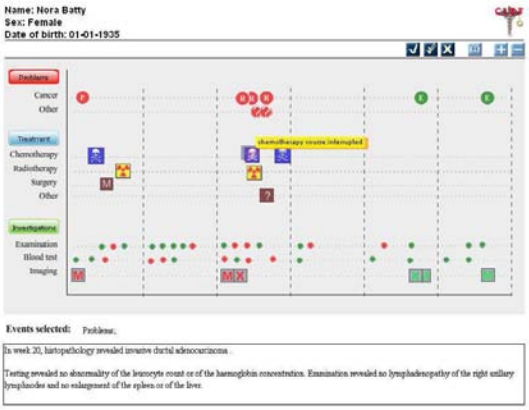


Figure 1: Visual history snapshot: minimum zooming, the user has selected *Problems* in the graphical interface and a summary of all recorded problems is displayed in the bottom pane.

The advantage of a visual display is that the user can have a global view of a patient's history. However, much information is hidden behind each event displayed on the timeline. The user can reveal this information by interacting with the graphical display. By zooming in or out, events are collapsed or expanded. For example, in Fig.1 there is a chemotherapy event spanning 8 weeks. In a minimum zoom view, they appear as a single chemotherapy event; by zooming in, the user will be able to see that there have been 6 chemotherapy cycles given succesfully, and 3 chemotherapy cycles have been deferred. Hovering the mouse over any event will display as a tooltip a short description of the event. For example, hovering over the chemotherapy event in Fig. 1, the user will see the tooltip *A complete chemotherapy course was given from week 312 to week 320.* Further information about an event can be obtained by clicking on its

icon. A chemotherapy event, for example, "hides" information about the particular drug regimen used, exact dates of chemotherapy cycles and reasons for deferring a particular cycle. Since this information is better expressed as text than graphically, each selection of an event will trigger the production of a report snippet that describes in more detail that particular event.

Apart from individual events, the user can also select multiple events (by clicking on several event icons on the timeline), classes of events (by clicking on the event name on the left hand side of the screen) or time spans (by selecting years on the horizontal axis). The effect of such selections will be the production of summaries similar to those described in the previous section. Selection of events will produce event-focused summaries, whilst selection of time spans will produce longitudinal summaries for that particular span.

Semantic relations between events are displayed on demand, allowing the user to see the logical sequence of events (tracing, for example, the reason for performing a red packed cell transfusion to anaemia which was in turn caused by chemotherapy performed to treat cancer).

3 The Report Generator

In the following, the term *Report Generator* will be used to designate the software that performs text generation, as a result of either a direct request from the user for a specific type of report or a selection of events in the graphical timeline. The output of the report generator may be either a full report or a report snippet, but practically, the type of selection employed in chosing the focus of the report does not influence the technique used in generating it.

3.1 Input

The input to the Report Generator is a chronicle, which is a highly structured representation of an Electronic Patient Record, in the form of a semantic network. It is beyond the scope of this paper to describe the methodology involved in transforming an EPR into a chronicle - the chronicalisation process is complex and involves Information Extraction from narratives, solving multi-document coreference, temporal abstraction and inferencing over both structured and information extraction data (Harkema et al., 2005). For the purpose of this paper, we consider the input correct and complete. The main advantage in using a chronicle as opposed to a less structured Electronic Patient Record lies in the richness of information provided. Having access to not only facts, but to the relations between them, has important implications in the design of the content selection and text structuring stages. This facilitates better and easier text generation and allows for a higher degree of flexibility of the generated text.

The chronicle relations can be categorised into three types according to their role in the generation process. *Rhetorical relations* are relations of causality and consequence between two facts (such as, `Problem` CAUSED-BY `Intervention` or `Intervention` INDICATED-BY `Problem`) and are used in the document planning stage for automatically inferring the rhetorical structure of the text, as it will be described in 3.2.2. *Ontological relations* such as `Intervention` IS-SUBPART-OF `Intervention` bear no significance in text planning and realisation, but can be used in content selection. *Attribute relations* such as `Problem` HAS-LOCUS `Locus` or `Investigation` HAS-INDICATION `Problem` are used in grouping messages in a coherent way, prior to the construction of the rhetorical structure tree.

3.2 System design

The system design of the Report Generator follows a classical NLG pipeline architecture, with a Content Selector, Content Planner and Syntactic Realiser. The Content Planner is tightly coupled with the Content Selector, since part of the document structure is already decided in the event selection phase. Aggregation is mostly conceptual rather than syntactic, therefore it is performed in the content planning stage as well.

3.2.1 Content selection

The process of content selection is driven by two parameters: the type of summary and the length of summary. We define the concept of *summary spine* to represent a list of concepts that are essential to the building of the summary. For example, in a summary of the diagnoses, all events of type `Problem` will be part of the spine (Figure 2). Events linked to the spine through some kind of relation may or may not be included in the summary, depending on the specified type and length of the summary. The design of the system does not restrict the spine to containing only events of the same type: a spine may contain, for example, `Problems` of type *cancer*, `Investigations` of type *x-ray* and `Interventions` of type *surgery*.

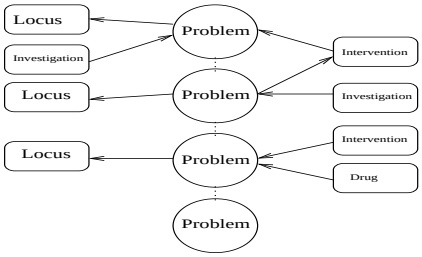


Figure 2: Spine events for a summary of diagnoses

The relations stored in the chronicle help in the construction of clusters of related events. A typical cluster may represent, for example, that a patient diagnosed with cancer following a clinical examination, a mastectomy was performed to remove the tumour, a histopathological investigation on the removed tumour confirmed the cancer, radiotherapy was given to treat the cancer, which caused an ulcer that was then treated with some drug. Smaller clusters are generally not related to the main thread of events, therefore the first step in the summarisation process is to remove small clusters[2] The next step is the selection of important events, as defined by the type of summary. Each cluster of events is a strongly connected graph, with some nodes representing spine events. For each cluster, the spine events are selected, together with all nodes that are at a distance of less than *n* from spine events, where *n* is a user-defined parameter used to adjust the size of the summary. For example, in the cluster presented in figure 3, assuming a depth value of 1, the content selector will choose *cancer*, *left breast* and *radiotherapy* but not *radiotherapy cycle*, nor *ulcer*.

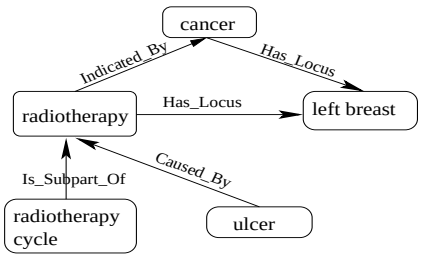


Figure 3: Example of cluster

The result of the content selection phase is a list of messages, each describing an event with some of its attributes specified. The number of attributes specified depends on the depth level of a message (i.e., how far from the spine the event is). For example, a `Problem` event has a large number of attributes, consisting of `name, status, existence, number of nodes counted, number of nodes involved, clinical course, tumour size, genotype, grade, tumour marker` and `histology`, along with the usual time stamp. If the *Problem* is a spine event, all these attribues will be specified, whilst if the *Problem* is 2 levels away from the spine, only the name and the existence will be specified.

[2]In the current implementation these are defined as clusters containing at most three events. This threshold was set following empirical evidence.

3.2.2 Document planning

The document planner component is concerned with the construction of complete document plans, according to the type of summary and cohesive relations identified in the previous stage. The construction of document plans is, however, initiated in the content selection phase: content is selected according to the relations between events, which in turn informs the structure of the target text.

The document planner uses a combination of schemas and bottom-up approach. A report is typically formed of three parts: a schematic description of the patient's demographic information (e.g., name, age, gender); a two-sentence summary of the patient's record (presenting the time span of the illness, the number of consults the patient attended and the number of investigations and interventions performed); and the actual summary of the record produced from the events selected to be part of the content. In what follows, we will concentrate on this latter part.

The first stage in structuring the summary is to combine messages linked through attributive relations. This results in instances such as that shown in example (3), where a `Problem` message is combined with a `Locus` message to give rise to the composite message `Problem-Locus`.

In the second stage, messages are grouped according to specific rules, depending on the type of summary. For longitudinal summaries, the grouping rules will, for example, stipulate that events occurring within the same week should be grouped together, and further grouped into years. In event-specific summaries, patterns of similar events are first identified and then grouped according to the week(s) they occur in; for example, if in week 1 the patient was examined for enlargement of the liver and of the spleen with negative results and in week 2 the patient was again examined with the same results and

underwent a mastectomy, two groups of events will be constructed, leading to output such as:

(3) In weeks 1 and 2, examination of the abdomen revealed no enlargement of the liver or of the spleen.
In week 2, the patient underwent a mastectomy.

Within groups, messages are structured according to discourse relations that are either retrieved from the input database or automatically deduced by applying domain specific rules. At the moment, the input provides three types of rhetorical relation: Cause, Result and Sequence. The domain specific rules specify the ordering of messages, and always introduce a Sequence relation. An example of such a rule is that a histopathology event has to follow a biopsy event, if both of them are present and they start and end at the same time. These rules help building a partial rhetorical structure tree. Messages that are not connected in the tree are by default assumed to be in a List relation to other messages in the group, and their position is set arbitrarily. Such events are grouped together according to their type; for example all unconnected `Intervention` events, followed by all `Investigations`.

In producing multiple reports on the same patient from different perspectives, or of different types, we operate under the strong assumption that event-focussed reports should be organised in a way that emphasises the importance of the event in focus. From a document structure viewpoint, this equates to building rhetorical structures where the focus event (i.e., the spine event) is expressed in a nuclear unit, and skeleton events are preferably in sattelite units.

At the sentence level, spine events are assigned salient syntactic roles that allows them to be kept in focus. For example, a relation such as `Problem` CAUSED-BY `Intervention` is more likely to be expressed as:

> "The patient developed a `Problem` as a result of an `Intervention`."

when the focus is on `Problem` events, but as:

> "An `Intervention` caused a `Problem`."

when the focus is on `Interventions`.

This kind of variation reflects the different emphasis that is placed on spine events, although the wording in the actual report may be different. Rhetorical relations holding between simple event descriptions are most often realised as a single sentence (as in the examples above). Complex individual events are realised in individual clauses or sentences which are connected to other accompanying events through the appropriate rhetorical relation. Additionally, the number of attributes included in the description of a `Problem` is a decisive factor in realising an event as a phrase, a sentence or a group of sentences. In the following two examples, there are two `Problem` events (`cancer` and `lymphnode count`) linked through an `Investigation` event (`excision biopsy`), which is indicated by the first problem and has as a finding the second problem. In Example 4, the problems are first-mentioned spine events, while in Example 5, the problems are skeleton events (the cancer is a subsequent mention and the lymphnode count is a first mention), with the `Investigation` being the spine event.

(4) A 10mm, EGFR +ve, HER-2/neu +ve, oestrogen receptor positive cancer was found in the left breast (histology: invasive tubular adenocarcinoma). Consequently, an excision biopsy was performed which revealed no metastatic involvement in the five nodes sampled.
(5) An excision biopsy on the left breast was performed because of cancer. It revealed no metastatic involvement in the five nodes sampled.

As these examples show, the same basic rhetorical structure consisting of three leaf-nodes and two relations (CAUSE and CONSEQUENCE) is realised differently in a `Problem`-focussed report compared to an *Investigation*-based report. The conceptual reformulation is guided by the type of report, which in turn has consequences at the syntactic level.

3.2.3 Aggregation

The fluency of the generated text is enhanced by conceptual aggregation, performed on messages that share common properties. Simple aggregation rules state, for example, that two investigations with the same name and two different target loci can be collapsed into one investigation with two target loci. Consider, for example, a case where each clinical examination consists of examinations of the abdomen for enlargement of internal organs (liver and spleen) and examination of the lymphnodes. Thus, each clinical examination will typically consist of three independent Investigation events. If fully expanded, a description of the clinical examination may look like:

(6) • examination of the abdomen revealed no enlargement of the spleen
• examination of the abdomen revealed no enlargement of the liver
• examination of the axillary lymphnodes revealed no lymphadenopathy of the axillary nodes

With a first level of aggregation, this is reduced to:
(7) Examination revealed no enlargement of the spleen or of the liver and no lymphadenopathy of the axillary nodes.

However, even this last level of aggregation may be not enough, since clinical examinations are performed repeatedly and consist of the same types of investigation. We employ two strategies for further aggregating similar events. The first solution is to report only those events that deviate from the norm - for example, abnormal test results. The second, which leads to larger summaries, is to produce synthesised descriptions of events. In the case of clinical examinations for example, it can describe a sequence of investigations such as the one in Example 7 as *"The results of a clinical examination were normal"*, or, if the examination result deviates from the norm on a restricted numbers of parameters, as *"The results of clinical examination were normal, apart from an enlargement of the spleen"*.

4 Related work

Natural language generation has been used in the medical domain for various applications. For example: to generate drug leaflets (i.e., pill inserts) in multiple languages and styles (PILLS (Bouayad-Agha et al., 2002)), letters to patients to help them stop smoking (STOP (Reiter et al., 2003)), individualised patient-education brochures (MIGRANE (Buchanan et al., 1992)); HealthDoc (Hirst et al., 1997)); Piglit (Binsted et al., 1995)). There is also a body of work on the generation of summaries of patient records (e.g., (Afantenos et al., 2005), (Elhadad and McKeown, 2001)). This work, however, differs from ours in that they concentrate on the summarization of textual records, while we deal with summarization of data from Electronic Patient Records.

Most computer-based patient record management systems have simple generation facilities built-in, which produce simple text, normally consisting of unconnected sentences and thus lacking fluency. Natural language generation techniques have been applied in various reporting systems for generating telegraphic textual progress notes (Campbell et al., 1993), reports on radiographs (A. Abella, 1995), and bone scans (Bernauer et al., 1991) or post-operative briefings (M. Dalal, 1996).

The timeline method has been used extensively in visualising patient histories. The Lifelines project (Plaisant et al., 1998) provides a method for visualising and accessing personal histories by means of a graphical interface, and has been used for both patient records and legal case histories. TeleMed (Kilman and Forslund, 1997) gathers patient information from distributed databases and presents it in a Web interface as icons on a timeline; interaction with the icons provides access to more detailed descriptions of the individual pieces of information. Various authors describe the advantages of the timeline approach to visualising temporal data of the kind present in patient histories (Tufte and Kahn, 1983; Cousins and Kahn, 1991).

5 Conclusion

We presented in this paper an innovative approach to the problem of presenting and navigating through patient histories as a means of supporting clinical care and research. Our approach uses a visual naviagation tool combined with natural language generated reports. Although developed for the domain of cancer, the very same methods that we could be used with little adaptation effort for general health care; they are based on a general model of the main events in a patient's medical history that occur across all diseases and ailments: symptoms, diagnoses, investigations, treatments, side effects and outcomes. As such, we only make use of general medical knowledge, which applies to any medical sub-domain. The design of both the text generator and visual navigator is completely data-driven by the system input, the chronicle.

An important consequence of the use of chronicles as input is that our system does not require complex domain semantics, which has been regarded as one of the essential components of NLG systems (Reiter and Dale, 2000). This is partly because inferences that are normally required to combine and order facts in the generated summary have already been performed prior to the language generation process, and their results have been stored in the chronicle as relations between facts. Indeed, a key feature of our system is that — apart from the relations present in the input data — it does not use any kind of external domain knowledge in the process of content selection. The only domain specific rules used are in text organisation, specifically in

the ordering of messages; these are not essential, although they do improve the fluency of the text. Our lack of reliance on domain semantics is a clear advantage for the portability of the system to other domains. It is nevertheless true that more specific domain knowledge could improve the summarization process, for example, in deciding which events should be considered important (which will clearly vary from one medical area to the next). In our report generation system, this type of knowledge can be encoded as a set of external rules, whose application would not be essential to the system. These rules can be specified without interfering with the main application, and require no changes in previous code.

Current electronic patient records are human-friendly but highly impoverished from the point of view of systematic machine analysis and aggregation. By contrast, the ideal machine representation is far too complex to be human-friendly. Our research suggests that with the combined graphical and natural language geneneration approach we have described here, this complex machine representation can be made both relatively familiar, and friendly.

Acknowledgment

The work described in this paper is part of the Clinical E-Science Framework (CLEF) project, funded by the Medical Research Council grant G0100852 under the E-Science Initiative. We gratefully acknowledge the contribution of our clinical collaborators at the Royal Marsden and Royal Free hospitals, colleagues at the National Cancer Research Institute (NCRI) and NTRAC and to the CLEF industrial collaborators.

References

J. Starren A. Abella, J. Kender. 1995. Description generation of abnormal densities found in radiographs. In *Proceedings of the Symposium on Computer Applications in Medical Care (SCAMC)*, pages 542–546.

Stergos D. Afantenos, Vangelis Karkaletsis, and Panagiotis Stamatopoulos. 2005. Summarization from medical documents: A survey. *Artificial Intelligence in Medicine*, 33(2):157–177.

J. Bernauer, K. Gumrich, S. Kutz, P. Linder, and D.P. Pretschner. 1991. An interactive report generator for bone scan studies. In *Proceedings of the Symposium on Computer Applications in Medical Care (SCAMC)*, pages 858–860.

K. Binsted, A. Cawsey, and R.B. Jones. 1995. Generating personalised patient information using the medical record. In *Proceedings of Artificial Intelligence in Medicine Europe*, Pavia, Italy.

N. Bouayad-Agha, R. Power, D. Scott, and A. Belz. 2002. Pills: Multilingual generation of medical inf ormation documents with overlapping content. In *Proceedings of LREC 2002*, pages 2111–2114.

B.G. Buchanan, J.D. Moore, D.E. Forsythe, G. Carenini, and S. Ohlsson. 1992. Involving patients in health care: explanation in the clinical setting. In *Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care (SCAMC'92)*, pages 510–512.

K.E. Campbell, K. Wieckert, L.M. Fagan, and M.Musen. 1993. A computer-based tool for generation of progress notes. In *Proceedings of the Symposium on Computer Applications in Medical Care (SCAMC)*, pages 284–288.

Cousins and M. Kahn. 1991. The visual display of temporal information. In *Artificial Intelligence in Medicine*, pages 341–357.

N. Elhadad and K. McKeown. 2001. Towards generating patient specific summaries of medical articles. In *Proceedings of the Workshop on Automatic Summarization, NAACL 2001*, Pittsburg, USA.

H. Harkema, I. Roberts, R. Gaizauskas, and M. Hepple. 2005. Information extraction from clinical records. In *Proceedings of the 4th UK e-Science All Hands Meeting*, Nottingham, UK.

Graeme Hirst, Chrysanne DiMarco, Eduard H. Hovy, and K. Parsons. 1997. Authoring and generating health-education documents that are tailored to the needs of the individual patient. In *Proceedings of the 6th International Conference on User Modeling*, Italy.

David G. Kilman and David W. Forslund. 1997. An international collaboratory based on virtual patient records. *Communications of the ACM*, 40(8):110–117.

K.McKeown M. Dalal, S. Feiner. 1996. MAGIC: An experimental system for generating multimedia briefings about post-bypass patient status. *Journal of the American Medical Informatics Association*, pages 684–688.

C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman. 1998. Lifelines: Using visualization to enhance navigation and analysis of patient records. In *Proceedings of the 1998 American Medical Informatic Association Annual Fall Symposium*, pages 76–80.

Alan Rector, Jeremy Rogers, Adel Taweel, David Ingram, Dipak Kalra, Jo Milan, Robert Gaizauskas, Mark Hepple, Donia Scott, and Richard Power. 2003. Clef - joining up healthcare with clinical and post-genomic research. In *Second UK E-Science "All Hands Meeting"*, Nottingham, UK.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge University Press, New York, NY, USA.

E. Reiter, R. Robertson, and L.M. Osman. 2003. Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144:41–58.

E.R. Tufte and M. Kahn. 1983. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

*e*Science Simulation of Clinic Electrophysiology in 3D Human Atrium

Sanjay Kharche[1], Gunnar Seemann[2], Lee Margetts[3], Joanna Leng[3], Arun V Holden[4], and Henggui Zhang[1]

[1]School of Physics and Astronomy, University of Manchester, Manchester, M60 1QD, UK
[2]Institute of Biomedical Engineering, University of Karlsruhe (TH), Karlsruhe, Germany
[3]Directorate of Information Systems, University of Manchester, Manchester, M13 9PL, UK
[4]Institute of Cell Membrane and Systems Biology, University of Leeds, Leeds, LS1 9JT, UK

Abstract

Atrial fibrillation (AF) is a common cardiac disease with high rates of morbidity, leading to major personal and NHS costs. Computer modeling of AF using a detailed cellular model with realistic 3D anatomical geometry allows investigation of the underlying ionic mechanisms in far more detail than in a physiology laboratory. We have developed a 3D virtual human atrium that combines detailed cellular electrophysiology, including ion channel kinetics and homeostasis of ionic concentrations, with anatomical detail. The segmented anatomical structure and multi-variable nature of the system makes the 3D simulations of AF large and computationally intensive. The computational demands are such that a full problem solving environment requires access to resources of High Performance Computing (HPC), High Performance Visualization (HPV), remote data repositories and a backend infrastructure. This is a classic example of *e*Science and Grid-enabled computing. Initial work has been carried out using multiple processor machines with shared memory architectures. As spatial resolution of anatomical models increases, requirement of HPC resources is predicted to increase many-fold (~ 1 – 10 teraflops). Distributed computing is essential, both through massively parallel systems (a single supercomputer) and multiple parallel systems made accessible through the Grid.

1. Introduction

AF is a condition in which the upper chambers of the heart contract at a very high rate and in a disorganized manner. As a result, the pulse of AF patients is highly irregular. AF affects about 4% of the population over the age of 60 years. Mortality rate associated with AF has increased over the past two decades. In the US, the age standardized death rate (per 100,000 US population) increased from 28 to 70 in 1998 [1]. In the UK alone, around 500,000 patients are affected by AF and the mortality rate has doubled since 1993. AF also increases the risk of stroke and heart failure among other complications.

The electrophysiology of the atrium, as in other cardiac tissue, is extremely complex. The membrane potential is determined by the integrated actions of ionic channels, Na/Ca exchanger and Na/K pump currents. In addition, regulative mechanisms governing the homeostasis of intracellular calcium concentration also affects the membrane potential. Modeling such a system requires use of high order, stiff ordinary differential

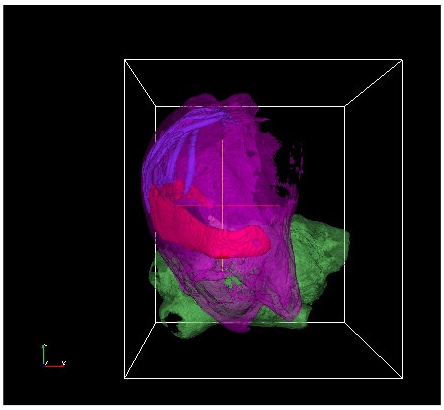


Figure 1. 3D anatomical model of human female atrium showing PM (blue), CT (red), RA (purple), LA (green). SAN and BB are embedded close to the CT and cannot be seen

equations. For example, the Courtemanche *et al.* [2] (CRN) model for human atrial cells consists of 24 state-variables of ODE. Such a description is necessary for the simulation of the effects of drugs, age and genetic disorders on ionic channel kinetics and thus single cell and tissue electrical activities. Simple models, *e.g.* FitzHugh-Nagumo (FHN) excitation model,

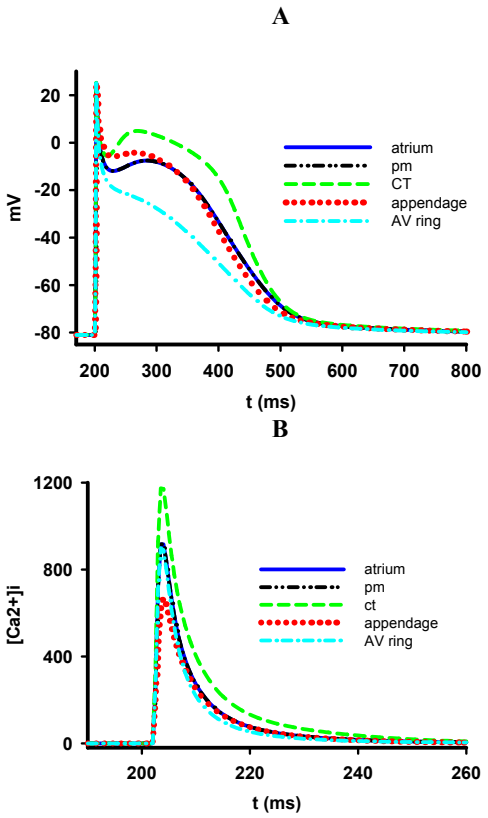


Figure 2. **(A)** AP heterogeneity in human atrium. LA, RA, PM and BB have an APD_{90} of 325 ms (solid blue), PM (dash dot-doted black), CT is 330 ms (dashed green). In the atrial appendage the APD_{90} is known to be 310 ms (dotted red), in the AV ring is 300 ms (dash dotted cyan). **(B)** Intracellular calcium concentrations associated with APs in **A**. Color coding and dash styles are as in **A**.

allow fast prototyping and (relatively) computationally inexpensive alternative for simulating electrical propagation in 3D geometries. Such models can be used as a starting point for the insertion of biophysically detailed cell models into anatomically detailed tissue models.

The human atrium consists of left atrium (LA), right atrium (RA), pectinate muscles (PM), cristae terminalis (CT), the Bachmann's bundles (BB), and sino-atrial node (SAN). Figure 1 shows a composite of all these components. The SAN is the pacemaker of the heart that initiates electrical action potential (AP). PM, CT and BB assist in conduction of electrical excitation waves. The human atrium is a heterogeneous tissue with cells in different regions having different action potential characteristics, preferential conduction pathways and varied fibre orientations.

2. Heterogenities in human atrial AP and anatomy

The CRN model is able to simulate AP in generic human atrial myocyte. However, the electrical activity in atrium is heterogeneous. The human atrium consists of several distinctively different regions with cells in each of the regions having different AP characteristics. The LA, RA, PM and BB have similar APs with an action potential duration (APD_{90}) of approximately 325 ms. However, in the CT the APD_{90} is 330 ms, in the atrial appendage the APD_{90} is known to be 310 ms, in the atrioventricular (AV) ring is 300 ms. The CRN model can be adapted by adjusting maximal conductances of the transient outward current (I_{to}), the L-type calcium current (I_{CaL}), and the rapidly activated potassium current (I_{Kr}) to simulate these heterogeneous APs [3] based on experimental data [4]. The results are shown in Figure 2, along with the corresponding heterogeneous calcium transients. Although there is heterogeneity in AP and calcium transients, the resting potential remains almost the same (-81.5 mV), as do the upstroke velocity (220 mV/ms) and the peak potentials (25 mV).

The SAN is a pacemaker, and is auto-rhythmic, with a period of approximately 70 per minute pacing the atrium at the base of the PM close to the CT. We have modified the CRN model by including a hyperpolarising-activated current (I_f) as given in Zhang *et al.* [5] to simulate the auto-rhythmic AP in SAN as shown in Figure 3. Details of modifications to the CRN model to simulate SAN AP are given in [3].

The 3D anatomical model for an adult female human atrium [6] was obtained from the National Library of Medicine of the National Institute of Health in Bethesda, Maryland, USA [7, 8]. This spatial resolution of the data set is much higher than that of clinical MRI. The voxel size is 0.33 mm x 0.33 mm x 0.33 mm. The data were segmented as described in [6].

In the atrium, fibre orientation is not as structured as in the ventricle. The fibre orientation of the atrial working myocardium has a complex and nearly random fashion [9]. Only the conduction pathways, *i.e.* CT, BB and PM and the tissue near the mitral and tricuspidal ostia, near the superior and inferior vena cavae and near the pulmonary veins have a consistent orientation. The macroscopic anisotropy of the electrical excitation conduction pattern, as well as mechanical contraction, is strongly influenced by the spatial layout of the cardio-

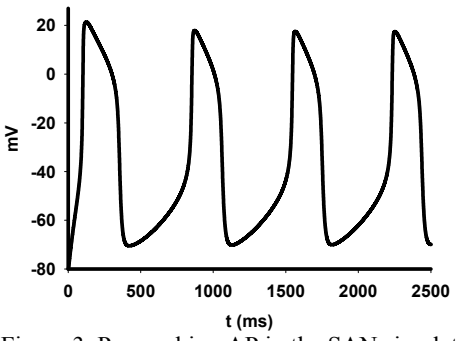


Figure 3. Pacemaking AP in the SAN simulated using modifications to the CRN model as given in [3].

myocytes, *i.e.*, the orientation of the muscle fibres and layers inside the myocardium. A nearly random fibre orientation is reported inside the human SAN [10]. These fibre orientations have been incorporated in our model.

Normal electrical excitation in the atrium has been reconstructed previously [5, 11]. Normal conduction starts at the SAN, which is the primary pacemaker of the heart with auto-rhythmic activity. Then the fast right atrial conduction pathways *i.e.* CT and PM transmit the excitation. During that phase, the right atrial working myocardium gets depolarised. At the same time, BBs transmit the activation towards the LA and the whole atrium is activated. All conduction pathways are characterized by a large electrical conductivity along the fibre direction. The excitation reaches the AV node, which is the only electrical path between atria and ventricles in the physiologically normal case. After a short delay, the excitation is then conducted into the ventricles. The simulated conduction pattern of atrial excitation is shown in Figure 4, using simple FHN cell model.

3. Computational resouces

The simulations were performed using a number of different computing facilities. These included systems hosted by the Computational Biology Laboratories (CBL) at Leeds, the University of Manchester and the National HPC Service, CSAR [12]. At CBL Leeds a 24 processor Sun-Fire-880 with UltraSPARC chip and a memory of 24 GB is available. The principal author's laboratory has a 4 processor Sun-Fire-880 with UltraSPARC chip with 16 GB of memory. A local university wide (University of Manchester) SGI machine with 32 SGI R14k processors and a memory of 16 GB is also available [13]. All these are shared memory systems (SMP). A distributed memory system with 16 dual processor Sun-Blade-1000 nodes is also available at CBL, Leeds.

CSAR provides, amongst other facilities, a 512 processor SGI Origin 3800 machine with 512 GB memory. Access to this system was granted through a Class 3 project, available for new users to gain access to the system for evaluation purposes.

The Sun-Fire machines have suitable up to date C/C++ compilers for parallelisation using OpenMP. MPI codes are compiled with MPICH (freely downloadable from the web) on the Sun machines. CSAR has MIPSPro 7 series and Intel v series of compilers.

4. Computing aspects of the 3D model – necessisity of HPC

The 3D anatomical atrial model, as described in the previous section, consists of 325 x 269 x 298 nodes. This amounts to more than 26 x 10^6 nodes. The following description of computing resource is based on implementation of the CRN 24 variable cell model, within human virtual atria. During a 3D simulation, the following arrays are required to be held in memory. A tissue information array (~10^8 integer values), the state array (10^9 consisting of double-precision floating-point data type), the diffusion matrix (10^9 double-precision floating-point data type), spatial derivatives are required to compute the heterogeneous conduction (10^8 double-precision floating-point data type). We can see that a minimum of 17.2 GB of memory is necessary. Instead of the CRN model, if a simple FHN model is used, the memory requirement is reduced to 10 GB due to a large reduction in number of independent variables associated with each node. In either case, large amounts of contiguous memory is required.

The serial code for 3D models with FHN as cell model was ran to estimate scalability. Simulating 1000 time units of activity took 5 hours of computer time on a standard Sun-Fire 880 workstation. Upon parallelization, the same run for FHN case was reduced to elapsed time of 1.5 hours using 4 processors, showing good scalability. Further optimization was implemented by use of binary output. This further improved performance by 7 % for the 1000 time unit simulation. Results of this simulation are shown in Figure 4.

Any pathology motivated case study requires many repeated simulations with different stimulus protocols or parameters in the model. This increases enormously the compute demand. AF due to re-entrant atrial excitation in

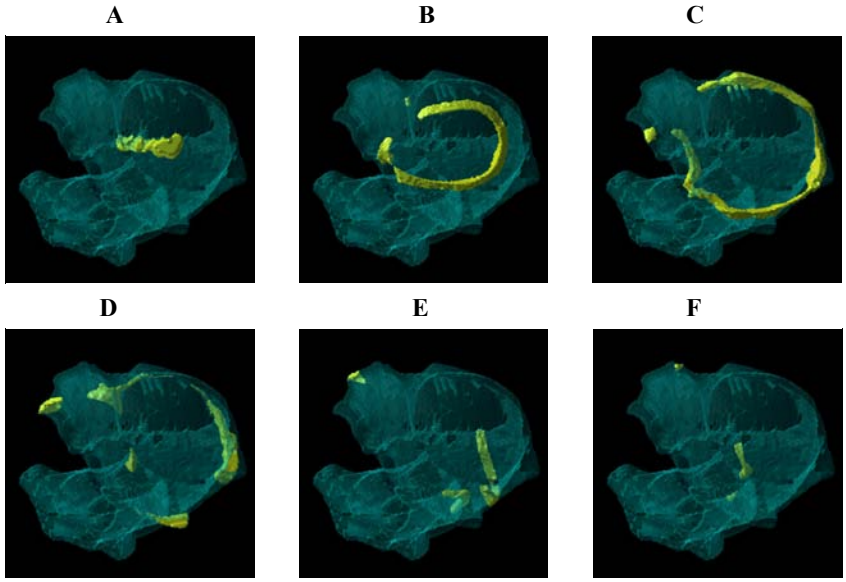


Figure 4. Simulation of normal conduction pattern in human female atrium. Translucent blue denoted the atrial geometry and solid yellow represents the excitation. Propagation is from the SAN to the LA. Model is stimulated at SAN (**A,** t = 0) and conduction spreads in the RA and along the BB and PM into the LA (**B, C, D, E, F**). Simulation of 1000 time units of activity (frame **F**) takes 95 minutes on a standard 4 processor Sun Sparc.

3D atrium is primarily propagating scroll waves. Scroll waves can be initiated in the atrium using several protocols. Simple S1-S2 or cut wave, or the following possible protocols to initiate re-entrant scroll waves are [14]:

1. S1 would be a stimulus at SAN. Then an ectopic excitation S2 located within the RA near the superior vena cavae, 15 mm away from the SAN.
2. Another protocol is the S1-S2-S3 protocol. The S1-S2 is as the same as described above, but after a time delay, a S3 stimulus is applied to the same location of the S2.
3. This protocol rapidly paces the SAN at high frequency. This results in re-entry on the RA surface after sufficient number of stimuli.

Initiation of re-entry at the required location in the 3D geometry using the S1-S2, S1-cut wave, or protocols 1 and 2 has to be done by trial and error, involving several trial simulations. Other possible stimulation protocols to induce normal and scroll waves have been described in [15, 16].

Upon using the cut wave protocol, we initiated re-entry and the results are shown in Figure 5. Such a simulation of 1 s takes about 44 hours on 16 processors using an improvement as described below.

Full geometrical model demands very large amounts of contiguous memory. We have, however, exploited problem specific features in our simulations and reduced these overheads considerably. Atrial tissue geometry occupies about 8% geometry of the total data set, due to atrium being thin walled, large holes of atrial chambers and vena caves. We re-structure (or renumber) the arrays mentioned in section 4 such that the real atrial nodes (leaving out the empty nodes) of the data set, *i.e.* only 8% of the total 26 million nodes and related information are stored. This improved efficacy of memory usage. By re-numbering the real atrial nodes we are not storing any data points that are not atrium. This reduced the memory required by FHN model to less than 3 GB. The memory required by CRN is reduced to less than 10 GB.

As a first step toward biophysically detailed modeling of human atrium, we have simulated electrical propagation using shared memory systems. A shared memory system is not necessary and the same results can be obtained using distributed memory systems. Distributed memory parallelism may give better scalability.

5. 2D Atrial and other small sized simulations

Often, 2D tissue simulations offer useful insights into the mechanisms underlying the

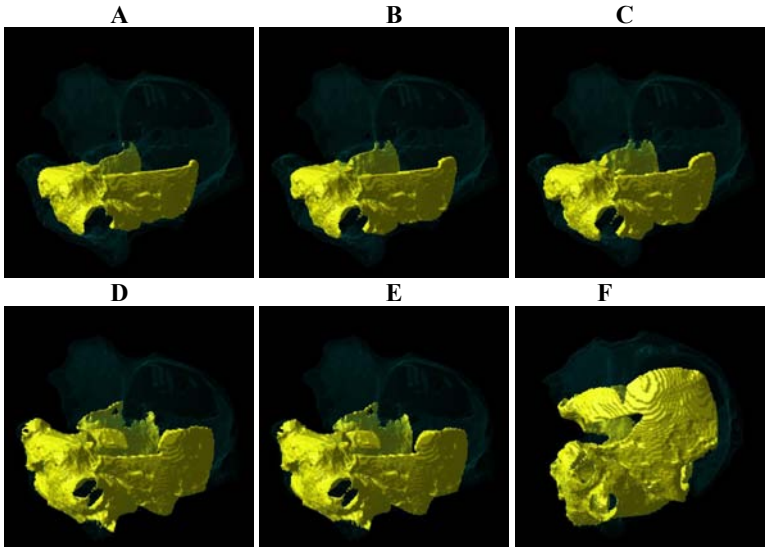


Fig 5. Initiation and propagation of scroll wave in human female atrium considering the healthy, or control case. Translucent blue shows atrial geometry, and solid yellow denotes propagating electrical activity. The SAN was stimulated to induce a solitary propagation. After an appropriate duration of time, the upper half of the geometry was reset to resting conditions simulating the cut wave protocol (**A**, t = 210 ms). Propagation of scroll wave can be seen in **B** (t = 230 ms), **C** (t = 250 ms), **D** (t = 270 ms) and **E** (t = 290 ms). In control case, re-entrant scroll waves self terminate (**F**, t = 335 ms).

genesis of AF, without dealing with complex 3D simulations. 2D tissue simulations were performed to investigate the link between AF genesis and gain-of-function in I_{K1} due to Kir2.1 gene mutation [17]. In this simulation, tissue size was taken to be 37.5 mm x 37.5 mm, in accordance with normal size of atrial appendage. Spiral re-entry was initiated with a standard cross-field stimulation protocol [18]. To characterize the stability of reentry, a 10 s long run of simulations were performed for 3 cases, consisting of control, for heterozygous and homozygous Kir2.1 gene mutation. Surface potentials was taken, to allow reconstruction of animation of activity from t = 0 to t = 10 s. In addition, pseudo-ECG and spiral wave tip positions were computed at run time. Sample frames from the 2D simulation results are shown in Figure 6.

Parallelization helps to reduce computing time and allows for a more intensive investigation. The program was parallelized using OpenMP and was ran on a dual processor Sun workstation. The time taken was 29 hours. We then ran it on 4 processor Sun-Fire and the elapsed time reduced to 16 hours. Our 2D code shows good scalability with near ideal speedup. The small fraction of code that has to be necessarily serial is while doing essential file output.

Cell models need to be investigated for various behaviors [19]. In cell models, a generic investigation is that of pacing based behavior of AP. Detailed of biophysical cell models in themselves are not overly demanding on memory, requiring storage of several tens or few hundreds of variables (*e.g.* models incorporating Markov chain formulations of cellular processes). Pacing based investigations are however, long by the nature of the problem. A ventricular cell model developed by Priebe-Beckmann (PB) [20] for non-failing and failing heart was pacing at various pacing rates from basic cycle length (BCL) = 100 ms, to BCL = 1200 ms in increments of 5 ms with trains of 100 stimuli. AP behavior for the final 10 stimuli were noted and plotted against BCL as a bifurcation diagram, to investigate the existence of AP alternans or 2:2 responses. Figure 7 shows the results from such a simulation. Simulation carried out on single processor as a serial run involved running this model for 60 hours. Inter-process communication in such a simulation is minimal, with the only synchronization required being in accumulation of the results. The pacing at a given BCL is independent of pacing result at another BCL. This being a cell model, the simulation is not demanding for memory of each node in the distributed memory system. Shared memory systems with large amount of memory are not required. If the code is parallelized for a

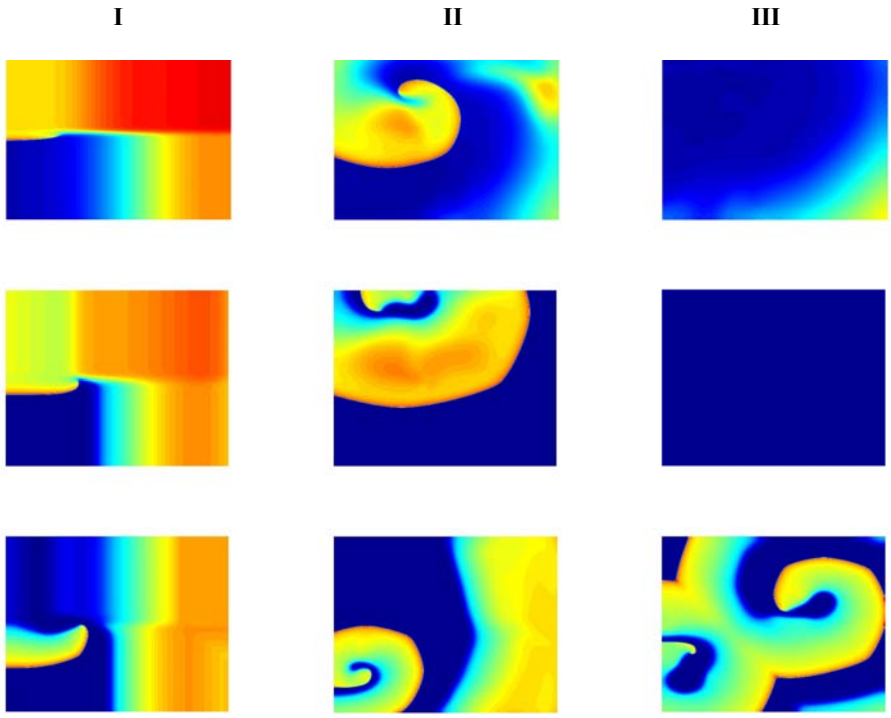


Figure 6. Representative frames from 2D simulations while investigating the effects of gain-of-function in I_{K1} on AF. Top panel shows frames for control, middle panel for heterozygous, and bottom panel for homozygous mutant type. 2D simulations at t = 400 ms in column **I**, t = 1300 ms in column **II**, and t = 2650 ms in column **III.** A total of 10 s of activity was simulated.

distributed memory system, then run time can be reduced drastically, depending on the number of nodes.

Another example where low cost distributed memory systems can be utilized is shown in Figure 8. Here we have a 1D transmural strand of human ventricle of length 15 mm with a spatial resolution of 0.1 mm with 150 cells, of which 40 are endo-, 50 are mid-, and 60 are epi-. Computing vulnerability window (VW) at a ventricular cell location requires running the 1D serial program on 1 processor for 1.3 hours. Again, as in the cell model case, computation of VW at a cell location is independent of computation of VW at another cell location. The memory requirement by the 1D model is moderate and can be well managed by nodes of any reasonable distributed memory system. Since VW computations at each of the cell locations are independent of each other, synchronization is not required. This problem also lends itself distributed memory parallelism.

6.Conclusions and Discussions

In this study, we have developed a 3D computer model of human atrium with detailed anatomical structures and cellular electrophysiology. This model provides an alternative to experimental methods to investigate the cellular ionic and molecular mechanisms underlying the genesis and control of AF in humans. Due to large size of geometry and multi-variable nature of the system, the model demands extensive computing resources

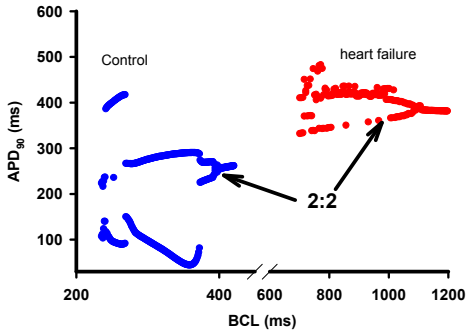


Figure 7. APD_{90} alternans (2:2 response) in PB model. Blue denotes control, red denotes heart failure. PB model is paced at BCLs from 100 ms to 1200 ms in increments of 5 ms. Dynamic response of last 10 APs from a train of 100 was noted. Alternans occur at low BCL.

and is an ideal test bed for HPC algorithms.

Simulations with low memory demands, but require long computations due to nature of problem are ideal for distributed computing.

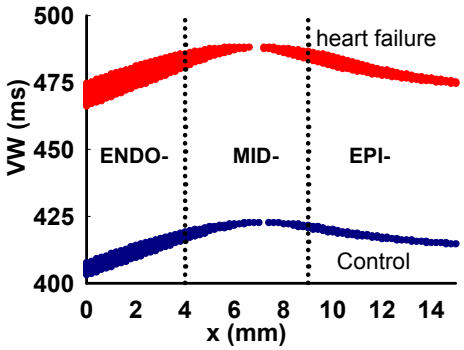


Fig 8. Computation of VW at each cell location of a 15 mm human virtual ventricular 1D strand with endo-, mid-, and epi- regions. A serial run of the program takes 200 hours. A single determination of VW takes 1.33 hours.

Acknowledgements

This work was supported by British Heart Foundation (PG/03/140) and BBSRC (BBS/B/1678X) UK.

References

[1] Wattingney WA, Mensah GA, Croft JB. Increased Atrial Fibrillation mortality: United States, 1980 – 1998. *Am J Epidemiol.* 2002; **155**: 819 – 826.

[2] Courtemanche M, Ramirez RJ, Nattel S. Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model. *Am J Physiol.* 1998 Jul; **275 (1 Pt 2)**: H301-21.

[3] Seemann G. Modeling of Electrophysiology and Tension Development in the Human Heart. 2005; Ph.D. Thesis. University of Karlsruhe.

[4] Feng J, Yue L, Wang Z, and Nattel S. Ionic mechanisms of regional action potential heerogeniety in the canine right atrium. *Circ. Res.* 1998; **83**: 541-551.

[5] Zhang H, Holden AV, Kodama I, Honjo H, Lei M, Vagues T, and Boyett MR. Mathematical models of action potentials in the periphery and centre of the rabbit sinoatrial node. *J. Am. Physiol.* 2000; **279 (1)**: H397-H421.

[6] Seemann G, Hoper C, Sachse FB, Dossel O, Holden AV, Zhang H. 3D anatomical and electrophysiological model of human sinoatrial node and atria. *Phil. Trans. Roy. Soc.*, 2006. In press.

[7] Ackerman MJ. "Viewpoint: The Visible Human Project". *J. Biocommunications.* **18 (2)**: 14 1991.

[8] "Visible Human Project, National Library of Medicine, Bethsda, USA".

[9] Boineau JP, Canavan TE, Schuessler RB, Cain ME , Corr PB, Cox JL. Demonstration of a widely distributed atrial pacemaker complex in the human heart. *Circ.* 1988; **77**: 1221-1237.

[10] Anderson RH, Ho SY. The architecture of the sinus node, the atrioventricular conduction axis, and the internodal atrial myocardium. *J. Cardiovasc. Electrophysiol.* 1998; **9:** 1233-1248.

[11] Harrild DM, Henriquez CS. A computer model of normal conduction in the human atria. 2000; *Circ. Res.* **87 (7)**: e25-e36.

[12] http://www.csar.cfs.ac.uk/

[13] http://www.mc.manchester.ac.uk/services

[14] Virag N, Jacquemet V, Henriquez CS, Zozor S, Blanc O, Vesin J-M, Pruvot E, Kappenberger L. Study of atrial arrhythmias in a computer model based on magnetic resonance images of human atria. 2000; *Chaos* **12 (3)**: 754 – 763.

[15] Vigmond EJ, Ruckdeschel R, Trayanova N. Re-entry in a morphologically realistic atrial model. 2001; *J. Cardiovas. Electrophysiol.* **12 (9)**: 1046 – 1054

[16] Haissaguerre M, Jais P, Shah DC, Takahashi A, Hocini M, Quiniou G, Garrigue S, Le Mouroux A, Le Metayer P, Clementy J. Spontaneous Initiation of Atrial Fibrillation by Ectopic Beats Originating in the Pulmonary Veins. *N Engl J Med*. 1998; **339 (10)**: 659-666

[17] Kharche S, Moore H, Garratt CJ, Hancox JC, Zhang H. Gain-of-Function in Kir2.1 and its Effects on Atrial Fibrillation in Homogeneous Virtual Human Atrial Tissue: A Computer Simulation Study. Poster at the ISCE 31st Meeting, Niagara-on-the-Lake, Canada. April 2006.

[18] Biktasheva IV, Biktashev VN, and Holden AV. Wave-breaks and self-termination of spiral waves in a model of human atrial tissue. *LNCS* 2005; **3504**: 293-303.

[19] Kharche S, Zhang H, Holden AV. Vulnerability in a One-Dimensional Transmural Model of Human Ventricular Tissue in Heart Failure. *Computers in Cardiology Conference, Lyon,* 2005; **32**: 563-566

[20] Priebe L and Beuckelmann DJ. Simulation Study of Cellular Electric Properties in Heart Failure, *Circ Res.* 1998; **82 (11)**: 1206-1223.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Integrative Biology: Real science through e-Science

Sharon Lloyd[1], David Gavaghan[1] , David Boyd[1], Denis Noble[1] , Blanca Rodriguez[1] , Thushka Maharaj[1] , Martin Bishop[1] , Richard Clayton[2] , James Handley[3], Ken Brodlie[3] , Gernot Plank[4]

[1]Oxford University Computing Laboratory, Wolfson Building, Parks Rd, Oxford, UK. OX1 3QD.

[2] University of Sheffield

[3] University of Leeds

[4] University of Graz, Austria

Abstract

'eScience can be described as the large scale science that will increasingly be carried out through distributed global collaborations enabled by the Internet.' This describes the fundamental requirements of the Integrative Biology project to support a diverse user community to bring together scientists and technologists to develop a 'collaboratory' for heart and cancer modelling. This paper aims to describe the scientific progress being made through the use of the facilities developed or made available to collaborating scientists in the area of complex heart modelling and describes how this new approach is changing the way that they perform their research.

1. Introduction

The human heart is an incredibly complex organ. It beats approximately 100,000 times a day, pumping about 5,000 gallons of blood, and beats 2.5 billion times over a typical 70 year life. A heart beat is caused by a wave of electrical stimulation spreading across the heart and this electrical activity turning into the mechanical movements that pump blood around the human body. The electrical excitation originates in the heart's pacemaker, the sinus node, located in the right atria (smaller chambers sitting on top of the ventricles). With such a complex organ, it is not surprising that its function can sometimes go wrong and when it does the results can be catastrophic. It is estimated that in 2002, cardiovascular disease (CVD) caused 39% of deaths in the UK, and killed just under 238,000 people. Ischemia is a condition in which the blood flow (and thus oxygen) is restricted to a part of the body. Cardiac ischemia is the name for lack of blood flow and oxygen to the heart muscle. Ischemic heart disease is the term given to heart problems caused by narrowed heart arteries. When arteries are narrowed, less blood and oxygen reaches the heart muscle. This is also called coronary artery disease and coronary heart disease. This can ultimately lead to heart attack. When a heart is not functioning properly and the electrical activity throughout the heart becomes chaotic, the heart flutters instead of beating normally. This process is called ventricular fibrillation and unless defibrillation by timely application of a strong electric shock is applied, the patient dies within minutes. Fibrillation, when initiated in the ventricles, the main pumping chambers, leads to death within minutes. The administration of a strong electrical shock (defibrillation) is the only effective therapy to restore a normal rhythm). When fibrillation occurs in the atria, this is not immediately life threatening, but cause discomfort. However, when atrial fibrillation (AF) is maintained long enough to become chronic, death may ensue due to secondary effects. Blood clots may form in regions of blood stasis which can get into the circulatory system and cause a stroke or heart attack.

Whilst it is recognised that heart disease takes many forms and there are well recognised causes e.g. obesity, congenital faults, poor diet, often the treatment of heart disease and subsequently survival rates are less well known. Clearly there are little or no opportunities to benefit from emergency situations to determine better treatment regimes or explore efficacy rates, as the survival of the patient is of primary concern. With coronary heart disease and heart attacks so prevalent, understanding how this process happens is crucial to enable clinical staff to make informed decisions about patient diagnosis and treatment.

2. Heart Modelling – the Science

What is evident from clinical observations is that whilst we can determine some information from activity on the surface of the heart and from observations of how it is pumping our blood, it is difficult to know what is actually happening within the vessels and the tissues and with experimental work on heart function limited, there has long been recognized the need to use alternative methods to determine how the heart works, how it behaves in arrhythmogenesis as well as how to treat these diseases. Realistic computational models of the heart, sometimes referred to as "virtual heart simulators", are currently the only viable approach to allowing us to observe all parameters of interest at sufficient spatial and temporal resolution. This section aims to highlight a selection of the heart modeling research which is being undertaken in conjunction with Integrative Biology.

Forty years of research have yielded extensive experience of modelling the cellular activity [1], [2] commencing with the work of Professor Denis Noble in Oxford in 1960. Since then, researchers have been simulating how cells behave and how electrical activity in the heart can be determined using these cell models. Oxford and Auckland have worked on joint research activity to explore cardiac activity through cell and tissue models and the recently funded Wellcome Trust Heart Physiome project shows their commitment to this field. This project aims to build on existing models to help understand different mechanisms of arrythmias. A simple change in a computer model or input parameter can replicate a mutation in a channel protein, for example, and the impact of that change can be followed in the model of the whole organ. This approach is already showing promising results and providing an insight into why arrythmias that look very similar when recorded on an electrocardiogram (ECG) can have many possible underlying causes.

If we take the models used by Denis Noble, which form the basis for many areas of research into heart disease and its treatment, we see that his models focused on the discovery and modelling of potassium channels in heart cells and the electric current flow through a cell wall [3] . Models of calcium balance were made in the 1980s and these have progressed extensively since then to a high degree of physiological detail. During the 1990s, these cell models were utilised to model anatomically detailed tissue and organ anatomy. Supplementing this is an understanding of how the tissues in the heart shear and move with every heart beat and how blood flows through the heart when it is pumping as well as an understanding proton transport mechanisms as modelled by Professor Richard Vaughan-Jones, as protons affect the metabolic function of heart cells. Molecular modelling by Professor Mark Sansom supports the work on proton transport.

Treatment of ischemia and fibrillation is an area of interest in heart modelling as resuscitation techniques in hospitals require informed refinement to improve patient survival rates. It has been found through in-silico studies in Oxford and Tulane that ventricular anatomy and transmural heterogeneities determine the mechanisms of cardiac vulnerability to electric shocks and therefore of defibrillation failure. This knowledge is important for the design of new protocols for defibrillation, which could consist for example in the application of small-magnitude shocks to particular locations in the ventricles. This could result in a dramatic decrease in the defibrillation threshold (which could be defined as the minimum shock strength required to terminate the arrhythmogenic activity in the ventricles in order to restore sinus rhythm or "to reset the ventricles").

Collaboration between the Medical University of Graz in Austria and University of Calgary has resulted in the development of a cardiac modelling package called CARP (Cardiac Arrhythmia Research Package, http://carp.meduni-graz.at). CARP is a multipurpose cardiac bidomain simulator which has been successfully applied to study the induction of arrhythmias , the termination of arryhthmias by electrical shocks (defibrillation) and also for the development of novel micro-mapping techniques. CARP is optimized to perform well in both shared memory and distributed memory environments. Besides pure computing capabilities, CARP comprises also visualization tools (Meshalyzer) and mesh generation tools. In Sheffield, computer models have been developed to model the process of ventricular fibrillation and resulting sudden cardiac death, as well as electric wave re-entry and fibrillation when the heart becomes unstable. Complex visualisation techniques have been developed to aid in this process and allow the tracking of filaments (centre of a spiral wave) in 2D and 3D where re-entrant activity manifests itself in the form of spirals (2D) and waves (3D).

Optical mapping is a widely used experimental technique providing high-resolution measurements of cardiac electrical activation patterns on the epicardial surface through the use of voltage-sensitive fluorescent dyes. However, inherent to the mapping technique are blurring and distortion effects due to fluorescent photon scattering from the tissue depths. Researchers in Oxford, in collaboration with US partners have developed a mathematical model of the fluorescent signal that allows simulation of electrical recordings obtained from optical mapping experiments A bi-domain representation of electrical activity is combined with finite element solutions to the photon diffusion equation to simulate photon transport within cardiac tissue during both the excitation and emission processes. Importantly, these simulations are performed over anatomically-based finite element models of the rabbit ventricles, allowing for direct comparison between simulation and whole-heart experimental studies [4].

These various approaches show how individual research is progressing in different areas and how, through collaboration with partner sites, the skills of other scientists are being leveraged.

3. Heart Modelling – the computing challenge

Modelling the human heart in its entirety is beyond the capabilities of even the largest supercomputer. Even the smallest modelling activity has historically proved to be a challenge for a typical modeller who has developed and refined their models on their local desktop of laptop. Models would typically run for many days if not weeks, limiting the capacity of simulations achievable with existing infrastructures. Some laboratories have been lucky enough to have local clusters usable by scientists and if these scientists were lucky enough to have the support of those managing these infrastructures, they could benefit from the use of local services. Management of their data has, however, been ad hoc and often scientists lacked the facilities to manage data correctly with suitable metadata and provenance capture capability. Preliminary statistics showed that to run 2ms of electrophysiological excitation of a slab of tissue from the left ventricular wall, using the Noble 98 heart model would require 1 day to compute on HPCx. This was calculated using a mesh with 20,886 bilinear elements (spatial resolution 0.6mm) and over 40 timesteps of 0.05ms timestep.

Clearly this is a huge scientific task which will require the collaboration of many researchers, but the infrastructure needs will also be extensive, with the need for seamless and secure data repositories to manage both the input data and the generated results, for fast compute facilities with potentially high speed networks enabling fast communication between the facilities, and extensive visualisation tools, generating geometry details remotely, enabling the scientists to visualise the results real time. By working with scientists in the design of such an infrastructure and in the development of overarching virtual research environments to support this research process, Integrative Biology aims to develop an infrastructure which enables this science to achieve the anticipated results in finding the key to preventing and treating disease.

The computing services we are developing within the project allow researchers to target simulations at the most appropriate computer system depending on the resources and response needed. They provide data and metadata management facilities, using the Storage Resource Broker (from UCSD), for looking after the many datasets created in computational experiments, and visualisation tools for examining results and discussing these collaboratively within the consortium. An associated project, funded by the Joint Information Systems Committee, is developing a Virtual Research Environment that is embedding these services into a portal-based framework so they are easily usable by researchers who are not themselves computing experts. Present capability allows users to submit their models to NGS and HPCx through a portal interface or Matlab interface and manage their data and provenance information through the concept of an experiment. This has opened up a new way of working for our scientific collaborators and the following section shows some of the results of this grid enabled science.

4. Examples of improved science through e-science

The following sections will describe the results of Integrative Biology users who have leveraged the technical capability offered to them to perform their science in new ways. These in-

silico experiments have resulted in publications in renowned journals and conferences (not only heart modelling, but also electrophysiology and computer science).

4.1 Heart de-fibrillation on NGS

Dr Blanca Rodriguez and her collaborators in Oxford and Tulane have worked on how to apply electric shocks to the heart to restart it and investigating the affect of disease of the heart on its effectiveness. Their research has explored 'windows of vulnerability' regarding the timing of its application of the shocks when the efficacy is reduced. To do so, they have used a code called Memfem from Tulane from the US, which uses a finite element solver and 2 PDEs (partial differential equations) coupled with a system of ODEs (ordinary differential equation). Each small project has about 150 jobs and each job generates 250 output files (each 4Mb), which amounts to 150Gb per project.

Figure 1. Finite element mesh of the Tulane model (141 828 nodes) with realistic fibre architecture

The ventricles are paced at the apex and after 7 beats, shocks of several shock strengths are applied to the computer model at different timings. Researchers vary the length of time between the last paced beat and the time of shock application (i.e. coupling interval), as well as shock strength to obtain the area of vulnerability, which is a 2D grid that encompass the combinations of coupling interval and shock strength that lead to shock-induced arrhythmogenesis. Many of the runs are independent so can be run simultaneously. Having access to NGS and SRB has been a vital resource in conducting this research. Already Dr Rodriguez and her team have made some surprise findings. "During the first 10-15 minutes after a heart attack, the shock strength necessary to defibrillate the ischemic heart is similar as in a healthy heart. This is surprising, as there are lots of changes going on in the oxygen deprived tissue over that time" [5,6]. These results have been obtained for a particular size of oxygen-deprived region with a rabbit heart model only and that they may not apply to larger regions or to the human heart. But they're sufficiently intriguing to warrant further investigation. It is hoped that subsequent experiments will be able to quantify the time limit for re-starting a heart and thus improve patient treatment.

4.2 Transmural electrophysiological heterogeneities:

Experimental and theoretical studies in both isolated ventricular tissue [7], and single myocytes [8] have proved that transmural dispersion in action potential duration (APD) exists, which results from changes in ionic properties in the depth of the ventricular wall. In particular, three layers of functionally-different cell types have been identified, namely the epicardial, endocardial and midmyocardial layers. Experimental and theoretical studies have proved that transmural dispersion in APD in the ventricles, which varies with age [9] and animal species [9] may modulate the arrhythmogenic substrate and thus could alter defibrillation efficacy. However, the role of transmural heterogeneities in the mechanisms underlying defibrillation failure is unknown.

Thushka Maharaj and her collaborators used the sophisticated computer model of stimulation/defibrillation developed at Tulane University, to provide mechanistic insight into the role of transmural electrophysiological heterogeneities in cardiac vulnerability to electric shocks [10]. The insight provided by the simulations revealed that increased transmural dispersion in action potential duration within the left ventricular free wall resulted in an increase in the likelihood of arrhythmia induction in the heart. Thus, the inclusion of electrical heterogeneities resulted in an increase in cardiac vulnerability to electric shocks. These simulations, requiring extensive computing power, were ran on the UK NGS and data were stored in the NGS SRB.

4.3 Understanding Ventricular Fibrillation

Current research at Sheffield University supported by the Integrative Biology project is focusing on understanding the mechanisms that initiate and sustain ventricular fibrillation (VF). Computer-intensive simulations using whole ventricle detailed anatomy and biophysically detailed models of electrical excitability are run

on HPCx, and these build on simulations using simplified models that are run on local HPC resources including the White Rose Grid. A key aspect of this work is to relate the findings to clinical practice. Work on modelling the initiation of VF has already yielded information that could be used to identify patients at risk, and this is the basis of a pilot clinical study about to start in collaboration with clinical colleagues at the Northern General Hospital in Sheffield. Work on the mechanisms that sustain VF is also tightly meshed with clinical and experimental studies, and is one component of a wider project focusing on understanding VF in the human heart that also involves the Universities of Auckland, Utrecht, Oxford, and UCL.

The current version of SCAM (the Sheffield Cardiac Arrhythmia Model) is written in C, uses shared memory parallelism, and runs on a single frame of HPCx. The code has been optimised for the HPCx architecture, and ongoing development aims to further exploit the mixed mode parallelism capability of HPCx, so that simulations can be run across large numbers of frames. Results of simulations using SCAM show initiation and development of fibrillation in the ventricles (Figures 2 and 3).

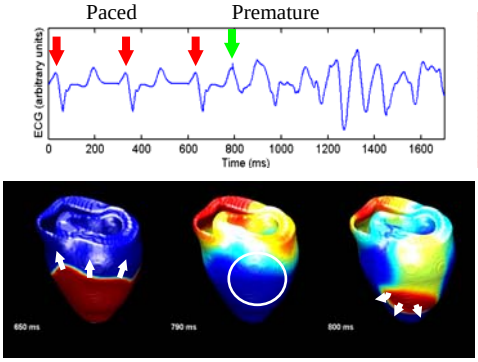


Figure 2. Simulation showing the initiation of fibrillation in the ventricles

The top panel shows simulated electrocardiogram (ECG). Red arrows indicate three stimuli to the apex (bottom) of the heart that result in normal placed beats at intervals of 300ms, and green arrow indicates premature stimulus delivered to the heart wall. The premature stimulus results in the onset of fibrillation, shown by rapid and self-sustained activity in the simulated ECG. Bottom panel shows snapshots of the simulation, where electrical activation is colour coded with blue indicating resting tissue and red indicating active tissue. The first frame (650 ms) shows the propagation of the third paced beat shortly after the stimulus has been applied to the apex (bottom) of the heart. The second frame (790 ms) shows the state of the heart just before the premature stimulus is applied over the region shown. Part of this region has yet to recover, shown by the yellow and light blue regions. The third frame shows the effect of the premature stimulus. The activity resulting from the stimulus can only propagate outwards and downwards because it cannot propagate into areas that are still recovering.

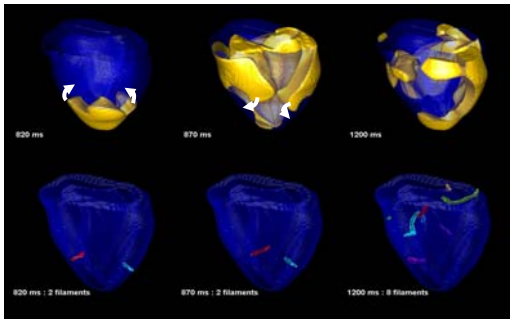


Figure 3. Simulation showing the development of fibrillation in the ventricles

Top panel shows isosurface views, where electrically active regions are enclosed by yellow surfaces. The first frame (820 ms) shows how the downward propagating activity shown in Figure 2 is beginning to curl around as the tissue recovers from the paced beat, and forms a pair of counter-rotating scroll waves. In the second frame (870 ms) these scroll waves have rotated by about 180 degrees, but the activation pattern is unstable, and by the third frame (1200 ms) the initial scroll wave pair has broken up into multiple interacting waves. Bottom panel shows scroll wave filaments – the lines around which the scroll waves rotate. In the first two frames there are two filaments, one for each of the scroll waves. In the third panel (1000 ms) there are 8 filaments, reflecting the more complex activation pattern resulting from the initial instability.

4.4 Cardiac Simulation on HPCx

In a preliminary study at the Medical University of Graz using the Integrative Biology framework, the feasibility of carrying out a "virtual experiment" was tested using HPCx. A computer model of a ventricle, discretized at an average spatial resolution of 200 μm, was simulated immersed in a conductive bath. At the

bath boundaries, two plate electrodes were placed next to the anterior and posterior faces of the ventricle (Figure 4A).To test conditions under which an arrhythmia can be induced, a train of 10 pacing pulses of varying basic cycle length was delivered (Figure 4B). After the last pacing pulse, 2 seconds of activity were simulated to examine whether an induced arrhythmia was sustained or self-terminated (Figure 4C).

Performing this virtual experiment involved the solution of an elliptic PDE (862,515 unknowns), a parabolic PDE (547,680 unknowns) and a set of 21 non-linear ODE's, defined at the same grid as the parabolic PDE. Using a temporal discretization step of 8µs, the solution scheme had to be repeated 500,000 times to complete the experiment. Preliminary simulations carried out on a Dual Opteron desktop computer suggested that execution times would be around 2 months. Using 128 CPUs of HPCx allowed the execution of a single experiment in only 10 hours. The simulations were carried out using the CARP simulator.

A subset of this simulation was repeated using different numbers of CPUs to demonstrate the scalability of the method (Figure 5). The overall computational workload is clearly dominated by the elliptic problem (> 95% of the overall workload). The parabolic PDE, solved by a simple forward Euler integration step, showed super-linear scaling. As expected, the ODE solver scaled linearly since the involved variables do not diffuse and thus no communication is required. The dominating elliptic problem scaled well, although the parallel efficiency decreased slightly when going from 64 to128 CPUs. Taking into account that this problem size is rather small for the high number of CPUs the scaling efficiency is more than satisfying.

These preliminary results suggest that realistic simulations of a human heart including a torso are feasible on the HPCx platform. In such simulations one has to deal with roughly 20-200 million unknowns (20-200 times larger than in this study). Memory usage and execution times will require the use of more CPUs. It is expected that parallel efficiency will increase significantly thanks to a more favourable ratio between local computational load and communication.

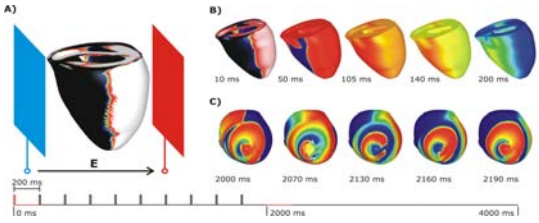


Figure 4. A virtual experiment.

Figure 4 shows results from a setup for a "virtual experiment" to induce an arrhythmia in the ventricles by applying an electrical pacing protocol: A) The ventricles are immersed in a conductive fluid and place between two plate electrodes. B) Electrical activation of the ventricles during the stimulation period. C) After the last pacing pulse, a so-called figure-of-eight re-entry (named after the movement of the tips of the wavefronts) ensued and was sustained until the end of the simulation run at 4000 ms

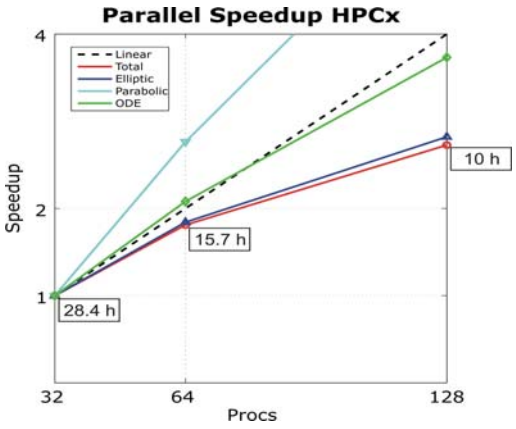


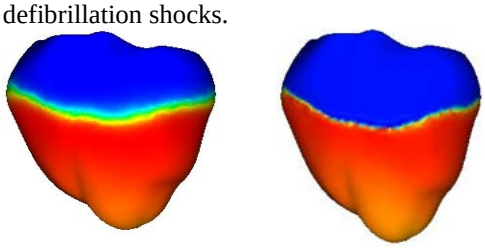
Figure 5. Benchmark results: scaling of different portions of the bi-domain computation

In figure 5, as expected, the ODE part scaled linearly (no communication required).The parabolic problem, solved by a simple forward Euler step, basically involved only a matrix-vector product which showed super-linear scaling. Computations were dominated by the elliptic problem which scaled reasonably well, particularly if one takes into account that the problem size is small for the number of CPU's employed in these simulations. Overall execution time in hours is shown as a function of the number of CPU's.

4.5 Modulation of Shock-End Virtual Electrode Polarisation as a Direct Result of 3D Fluorescent Photon Scattering

The following project has been conducted by Martin Bishop in Oxford using complex mathematical models from Tulane and results from experiments conducted by Igor Efimov's laboratory in St. Louis.

For the first time, a model of photon scattering has been used to accurately synthesize fluorescent signals over the irregular geometry of the rabbit ventricles following the application of strong defibrillation shocks. During such stimulation protocols there is a large transmural variation in transmembrane potential through the myocardial wall. It is thus thought that fluorescent photon scattering from depth plays a significant role in optical signal modulation at shock-end. A bidomain representation of electrical activity is combined with finite element solutions to the photon diffusion equation, simulating both the excitation and emission processes, over an anatomically-based model of ventricular geometry and fiber orientation. Photon scattering from within a 3D volume beneath the epicardial optical recording site is shown to transduce these differences in transmembrane potential within this volume through the myocardial wall. This leads directly to a significantly modulated optical signal response with respect to that predicted by the bidomain simulations, distorting epicardial virtual electrode polarization produced at shock-end. We can also show that this degree of distortion is very sensitive to the optical properties of the tissue, an important variable to consider during experimental mapping set-ups. These findings provide an essential first-step in aiding the interpretation of experimental optical mapping recordings following strong defibrillation shocks.



(A) **(B)**

Figure 6 Surface distribution of transmembrane potential (A) and synthesized optical signal (B) 50ms following apical stimulation.

5. Supporting Technology

Enabling this science is an architecture that abstracts away the complexity of working with disparate compute and data resources. This architecture has leveraged work from early e-Science projects. In order to handle the diversity of end-users – or to be more precise, the diversity of environments in which our end-users routinely work – and the diversity of IT systems which they wish to exploit, the infrastructure is organized into 4 major layers.

The upper layer is provided in two formats, one portal based to minimize the footprint on the users desktop/laptop, and the other based on a variety of familiar desktop tools such as Matlab to provide modellers with access to IB services in an environment in which they have already invested a significant learning and development effort. For some services, such as advanced visualization specific to these types of applications, some client-side support is required, necessitating the downloading of specific clients. This layer will also provide the interaction services for collaborative visualization which are required for steering and in situ decision making by partners.

The next layer contains Integrative Biology (IB) Services. These provide a defined, extensible set of services offering a simple abstract interface to the range of IT systems and facilities required by IB cardiac and tumour modellers. Supporting the IB Services are a set of Basic Services targeted at the various IT systems available to IB. These include front-ends to the Storage Resource Broker (SRB) data management service running on the UK National Grid Service (NGS) and low level job submission facilities provided by the Globus Toolkit 2 (GT2) and the Open Middleware Infrastructure Institute (OMII) , plus access to resources such as visualization and steering developed by the project. This Basic Services layer and the higher IB Services layer are Web service based. The Basic Services provide access to the underlying IT infrastructure systems used by the project through a variety of access mechanisms.

Management of the data created within a distributed research community is a major challenge for the IB project. Important factors are the provision of a secure infrastructure in which users can store their valuable and potentially sensitive data with the ability to control who has access to that information; the ability to share data selectively through the

concept of a collaborative experiment with the principal investigator in the experiment controlling access rights; and leveraging mature facilities for data management already developed by one of the project partners, CCLRC. The total volume of project data held in SRB is approaching 0.5TB, the majority of which is raw output from numerical simulations. IB files are organized by studies and experiments and managed through use of a metadata schema based on the CCLRC Schema for Scientific Metadata.

6. Conclusions

Researchers across the Integrative Biology consortium are developing complex computer models to look at aspects of how the heart operates and by coupling these models, these researchers hope to be able to determine what causes heart disease and potentially how to prevent heart disease. These complex models require immense amounts of compute power which is not available on the scientists desktop. For this reason, the project is enabling access to HPC resources through the National Grid Service (NGS) and High Performance Computing Service (HPCx) by utilising tools to enable the complexity of working with such infrastructures to be hidden from the scientist. This infrastructure and services are enabling modellers to do far more than build static models of the heart. By having almost instantaneous access to data, computing and visualisation software held on the NGS and HPCx, and ultimately global IT infrastructures, the IB project hopes to allow researchers to change conditions in the heart and see how the model reacts in almost real time – the nearest thing to experimenting with a live heart. Early feedback from the users has shown that by partnering with providers of advanced technology facilities, science can progress faster and the results shown here are evidence of the advances made to date in benefiting clinical patient care.

7. Acknowledgements

We wish to thank other members of the Integrative Biology consortium for their input into, and comments on, the work described in this paper, in particular the role of the scientific collaborators in determining the requirements for the project and providing the results for this paper. We also wish to thank the EPCC for help with installing and optimising our codes on HPCx. We acknowledge the financial support of the EPSRC (ref no: GR/S72023/01) and IBM for the Integrative Biology project, and of the Joint Information Systems Committee for the Integrative Biology Virtual Research Environment project. R. Clayton acknowledges support from the British Heart Foundation (PG03/102/15852). G. Plank was supported by the Austrian Science Fund FWF (R21-N04).

8. References

[1] Noble D. Modelling the heart: from genes to cells to the whole organ *Science* 2002; 295: 1678-1682.

[2] Noble D. The Initiation of the Heartbeat. OUP, Oxford. 1975

[3] A Modification of the Hodgkin-Huxley Equations Applicable to Purkinje Fibre Action and Pace-maker Potentials, Noble, D. 1962. *Journal of Physiology* 160, 317-352. PubMed ID: 14480151

[4] Synthesis of Voltage-Sensitive Optical Signals: Application to Panoramic Optical Mapping, Martin J. Bishop , Blanca Rodriguez , James Eason , Jonathan P. Whiteley , Natalia Trayanova and David J**.** Gavaghan *Biophys J.* BioFAST on January 27, 2006.doi:10.1529/biophysj.105.076505

[5] Role of shock timing in cardiac vulnerability to electric shocks, Rodríguez B, Trayanova N, Gavaghan D. . APICE International Symposium on Critical Care Medicine, Trieste (Italy), November, 2005.

[6] Vulnerability to electric shocks in regional ischemia, Rodríguez B, Tice B, Blake R, Eason J, Gavaghan D, Trayanova N. . Heart Rhythm, May 2006.

[7] C. Antzelevitch, G.X. Yan, W. Shimizu, and A. Burashnikov, "Electrical heterogeneity, the ECG, and cardiac arrhythmias, From Cell to Bedside," W.B. Saunders Company, Philadelphia, 1999.

[8] MA McIntosh, SM Cobbe, GL. Smith, "Heterogeneous changes in action potential and intracellular Ca2+ in left ventricular myocyte sub-types from rabbits with heart failure," *Cardiovasc Res.*, pp. 397-409, 2000.

[9] S.F. Idriss and P. D. Wolf, "Transmural

action potential repolarisation heterogeneity develops postnatally in the rabbit," *J. Cardiovas Electrophysiol*, pp. 795-801, 2004.

[10] T. Maharaj, B. Rodriguez, R. Blake, N. Trayanova, D. Gavaghan. Role of transmural heterogeneities in cardiac vulnerability to electric shocks. Heart Rhythm, 2006.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Developing an Integrative Platform for Cancer Research: a Requirements Engineering Perspective

Vito Perrone[1], Anthony Finkelstein[1], Leah Goldin[1], Jeff Kramer[2], Helen Parkinson[3,4], Fiona Reddington[3]

1 University College London, London UK
2 Imperial College, London UK
3 NCRI Informatics Coordination Unit, London UK
4 European Bioinformatics Institute, Cambridge UK

Abstract

The NCRI Informatics Initiative has been established with the goal of using informatics to maximise the impact of cancer research. A clear foundation to achieving this goal is to enable the development of an informatics platform in the UK that facilitates access to, and movement of, data generated from research funded by NCRI Partner organisations, across the spectrum from genomics to clinical trials. To assure the success of such a system, an initial project has been defined to establish and document the requirements for the platform and to construct and validate the key information models around which the platform will be built. The platform will need to leverage many projects, tools and resources including those generated by many e-Science projects. It also required contributing to the development of a global platform through a close interaction with similar efforts being developed by the NCI in the USA. This paper recounts our experience in analysing the requirements for the platform, and explains the customised analysis approach and techniques utilised in the project.

1. Introduction

A critical factor in the advancement of research in any e-science domain is the ease with which data can be integrated, redistributed, and made accessible and analyzable. This is particularly true in the cancer research domain where tackling these challenges is key to integrating diverse scientific data to inform clinical decision-making and enabling a move towards personalised medicine. A number of factors have so far hampered the ability of researchers to achieve interoperability across the various data sets in the cancer domain and to provide effective access to integrated data and services. Data sets have been generated by different research groups around the world working across the cancer research spectrum, that is, from basic to clinical cancer research. These distributed and heterogeneous data sets have been recorded and often made accessible (typically via web sites) in non-standardized ways, that is, using different vocabularies, data structures, metadata standards and service interfaces. In recent years, the need for standards has been recognized and a number of projects and initiatives have been established to define reference ontologies (within specific sub-domains or across sub-domains), common metadata elements, data representation formats, common data repositories, and so forth. Currently there are many informatics projects, resources and standards in use in the UK and internationally. Many of these are excellent research tools, but they have evolved separately and so present an incoherent, fragmented landscape. Furthermore, it is unclear whether these resources and standards have responded more to technical imperatives than user requirements.

1.1 The NCRI Informatics Platform

In this context, the goal of the NCRI Informatics Initiative (hereafter NCRIII) [1] is to increase the impact of UK cancer research and improve prevention and treatment of cancer by effective use of informatics to manage and exploit the vast amounts of diverse information currently generated. The envisaged support will be provided through an integrative platform [16] whose main aim is to enable the creation of an open community where the different informatics

tools and resources available in the UK and worldwide can interoperate with one another as a coherent whole. This will reduce duplication of effort and funding and leverage existing resources for maximum benefit.

1.2 The NCRIII Platform Requirements Analysis Project

As with any complex and innovative software system project, a fundamental activity is to reach a clear understanding of what the system's goals and requirements are. Lack of proper requirements analysis is recognized as the major source (almost 50%) of failure for software system projects [2]. E-science projects are, we would contend, not immune to this. A proper requirements analysis should: identify what is missing in the context where the system will operate, define stakeholder goals and expectations; eliminate ambiguities, inconsistencies and problems potentially leading to late faults; provide the information needed to support the system construction and the evaluation of competing technical options; enable system change and evolution.

Within the NCRIII the need for a precise identification of the key goals and requirements has been considered of primary importance due to the complexity of the envisioned project and the apparent lack of solutions capable of filling the existing gaps. The platform development has been thus preceded by a project focusing on the requirements analysis. A multidisciplinary analysis group has been set up and the analysis has been driven by a set of use cases acquired by interviewing practitioners working in the field. Unlike use cases collected in other e-science project, which describe the main functionalities of the system being developed, our use cases needed to uncover interoperability needs. The main goals of our analysis have been to understand how the various initiatives operating in the cancer research field would effectively cooperate with one another, what the relative roles are, how they are actually used by practitioners, and what may make the community self-sustainable. This has required us to assume a higher level perspective in defining the use cases which are closer to user stories as defined in the Agile development [22] than to standard use cases as described in [3]. The core characteristic of our project along with the intrinsic multidisciplinary nature of the project team and the need to work side by side with domain experts, have required us to adopt an innovative approach and to customize the used analysis techniques. In this paper we describe the approach and the used techniques, motivating them with respect to the project and domain characteristics.

Although our approach has been developed for supporting analysis of the cancer research domain, we believe that it has generic applicability to other projects in the broader context of e-science.

2. The Platform Context and Related Works

The context where the NCRI platform will operate is made up of a multitude of projects, resources and initiatives that aim to support cancer research in different ways. These range from local projects to global initiatives and address different issues ranging from data acquisition, storage, and mining to knowledge management. Furthermore, the field of cancer research is highly dynamic reflecting the emergence of new technologies, developing infrastructure and scientific discovery.

Our field analysis has led us to organize the different entities involved in the platform context into the following categories.

Local Repositories: belong to institutions or research groups and tend to host data about investigations performed by the local teams. They are important sources of data although the terminologies and data formats used to store the acquired data or specimens tend to be highly heterogeneous between repositories. Access to the data is usually limited to local researchers or exploiting direct contacts with them.

Specialized Global Repositories: aim to collect information about specific types of data e.g. ArrayExpress [4], and caArray [5], for micro-array data. Researchers need to be supported in both depositing and accessing the data they contain. Repositories may overlap with one another in the type of data they house and can use different data formats and metadata vocabularies to annotate the acquired data. Most of the repositories offer ad-hoc defined input forms for collecting data and a web site for accessing the stored data typically through a search engine. Other access services like API or Web Services may be offered.

Knowledge Bases: aim to collect domain knowledge in specific areas, organizing it so that deductive reasoning can be applied to them. Each repository focuses on specific data types and provides ad-hoc defined forms for input data and access functionalities which may include advanced user graphical inter interfaces. Examples include REACTOME (a curated KB of biological pathways) [6], PharmGKB (an integrated resource about how variation in human

genes leads to variation in our response to drugs) [7], UniProtKB/Swiss-Prot (a Protein Knowledgebase containing annotated protein sequences) [8].

Scientific Libraries: collect scientific publications containing research results in the field of interest. The basic information unit is a scientific paper. Examples include the Public Library of Science[9], and PubMed [10]

Policies: define the way information is collected and used .Examples include those provided by the Department of Health [12]

Terminology Sources (or ontologies): defined for offering terminology annotation services to the global cancer research community. They can vary in terms of addressed domains, coverage extension (domain specific vs. cross-domain), description language, adoption level, access service interfaces, etc. Examples include the NCI-thesaurus (cross-domain annotations in the cancer research domain)[5], the GO ontology (annotation of gene products), the MGED-ontology (descriptors for microarray experiments) [15], etc.

Representation Formats: defined to represent data for specific domains. Common representation formats are not available for all the cancer domains and often there are several proposals or "standards" in use. Examples include HL7 [13] and CDISC [14] for clinical information exchange, MAGE-ML [15] for microarray experiments, etc.

Bioinformatics Services: existing services used to elaborate raw data (e.g. microarray experiment data clustering), to search similarities between new biologic data and existing data stored in some repositories (e.g. BLAST), to translate data among different formats, etc.

Although not all existing projects, resources and initiatives can be easily categorized, this list provides a broad high-level view of the reference context showing its multiple facets.

The issue of interoperability is currently being addressed via initiatives underway at the US National Cancer Institute Center for Bioinformatics (NCICB) [5] and the European Bioinformatics Institute (EBI) [4]. NCICB is coordinating an ongoing project called the cancer Biomedical Informatics Grid (caBIG) whose aim is to maximize interoperability and integration among initiatives funded by the NCI by providing open source infrastructure and tools plus vocabulary services (EVS – Enterprise Vocabulary Services [5] and a centralized metadata repository (caDSR – Data Standards Repositories [5]). The EBI provides a centralized access point to several bioinformatics databases and services through a unique web site.

Although the ultimate goal is similar, improving research efforts by enabling interoperability, a number of differences exist between the envisioned NRCRIII platform and the existing initiatives [16]. Essentially, while the existing initiatives are addressing the problem by undertaking an approach whereby tools, vocabularies, common repositories, etc are built and access is centrally provided by the lead organization, the NCRI initiative aims to provide a 'glue' layer enabling the different initiatives, resources and tools to communicate with one another in a more effective fashion. This requires close collaboration with aforementioned organisations and other existing projects including CancerGRID [17], CLEF [18], etc.

3. The Analysis Approach

A feature of e-science projects is the need to establish a multidisciplinary team including requirements engineers, software developers and domain experts with a broad vision of the specific fields, in this case the cancer biomedical domain. Such a team was composed including requirements experts from University College London and Imperial College and domain experts from the NCRI Informatics Coordination Unit and Task Force. In the initial phase, cancer related literature analysis and periodic team meetings have been intertwined to build up a common understanding of the platform's high level goals and to define a common language. Outcomes of the preliminary activities have been a *context diagram* (whose main components have been described in outline in section 2), an initial set of *stakeholders* and a preliminary *domain model*. Subsequently, the analysis has been focused on understanding how the various elements operating in the platform context can interoperate one with another to underpin the needs of scientific researchers. In accordance with the NCRI vision of supporting community driven development, we have adopted a user centred approach for the analysis activities. To this end, a number of use cases, covering the whole spectrum of the cancer research as proposed by the NCRI matrix [1], have been acquired and used to drive the overall analysis process.

Figure 1 maps out the analysis flow focusing on the use case analysis.

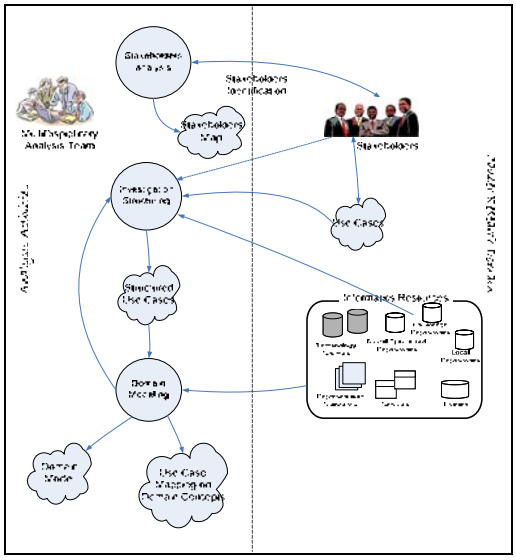


Figure 1: The use case analysis flow

Information acquired in the initial phase has been used to begin the stakeholders analysis activity. The method used, known as the "onion model" [20], allows a progressive identification and classification of the key stakeholders by means of interviews and other elicitation techniques. The bidirectional line in Figure 1 points out that the information acquired in each interview with stakeholders may lead to the identification of further stakeholders. Besides progressing in building the *stakeholders map* (available on [19]), in each interview with potential users a number of *high level use cases*, akin to "user stories" in the Agile development [22], have been defined. They describe stories on cancer research investigations that can benefit from the introduction of the platform. A simple example is shown in Figure 2. Working side by side with our experts for defining such stories, we have encouraged them to avoid thinking how the platform could satisfy their research needs and to instead carefully state their goals and what they would ask of the platform to achieve their goals. Such an approach permits uncovering of real users' needs and avoids potential biases, introduced by having in mind premature solutions [21].

Although rich in information, use cases acquired in this way need to be better structured to support further analysis. The **Investigation Structuring** activity, further described in section 4, aims thereafter to provide the use cases we acquired with a structure suitable for analysis. In summary, it enables the analysis team, and eventually the users involved in the use case definition, to systematically identify ***what*** the platform should be able to provide the researchers with in order to fulfil their goals.

Research Area: Functional Genomics
Actor: Researcher in a microarray laboratory
Description
A scientist wishes to *investigate genetic variation in tumour response to treatment with a specific class of chemotherapy*. She would like to *identify specimens of a specific tumour type, flash-frozen and prepared using a specific methodology, and for which there are associated medical records for treatment outcome.* With *sections of those specimens*, the researcher would like to *carry out microarray experiments for tumour cells and normal cells on the periphery of the tumour.* She needs to *store and analyze the data using conventional clustering methodologies.* She would also like to *compare the clusters to currently-known metabolic pathways, some of which are known to be chemotherapy targets.* With *the list of genes from the pathways of interest showing expression variation related to the chemotherapy treatment*, the investigator can then *identify common genetics variations* in the public databases for subsequent follow-up. At the time of publication of her study she wants to maximize the impact of her achievements on the scientific community for follow-up studies.

Figure 2: An example from the use case collection

The *structured use cases* are then analyzed in the **Domain Modelling** activity with two main purposes. Firstly, the *domain model* is validated, or modified, to reflect the use cases. Secondly, the actual resources available in the cancer context, described in terms of domain model concepts, are mapped to the use cases enabling a precise study of the interoperability issues. Investigation Structuring and domain modelling derive from well-known techniques in the requirements engineering field but have been customized to address the specific needs of this project. The next sections describe how they have been used, highlighting the aspects of a typical e-science project that have influenced the customization of these techniques.

4. Using an High Level Investigation Model in the Use Case Analysis

The use cases typically describe examples of investigations researchers perform in their daily work. The descriptions are completely unstructured and hard to understand for non-specialists like the computer scientists in charge of the analysis activities. An important step in their analysis is to structure them so that the investigation's goals, flow and the informatics resources needed by researchers can be clearly identified. To this end, we have defined a high level model, shown in Figure 3 enabling simple but effective structuring of investigations in terms of three primary concepts – Goals, Questions and Results (GQR) – and two subsidiary concepts – Services and Data Sets. The model is grounded by goal oriented requirements engineering principles [23] and is inspired by the Goal/Question/Metric method [24] used to plan the measurement of success of a software system from the user point of view.

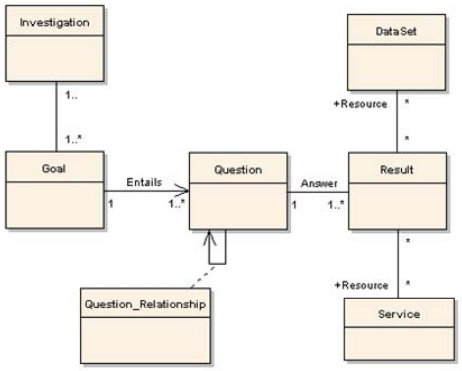


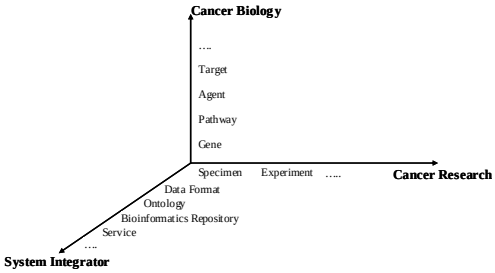
Figure 3: Core concepts in the GQR method

A **Goal** represents the investigation's objective. Goals can be more or less specific like "the role of diet in cancer" or "investigate whether a disease responds to a drug". A goal entails one or more **questions** that must be answered to achieve its fulfilment. Questions are answered by way of *data sets* or *services* (which in turn may consume data sets) building up the actual **results** the researcher was looking for. Data sets and services can be considered either inputs or outputs of an investigation. In the former case, one or more repositories or service providers that can offer the required data sets or services should be identified. In the latter case, repositories or service registries are used to publish data or services produced within the investigation. Results act as a bridge between the use case analysis and the domain modelling. They are used in combination with the domain model to identify what domain entities represent the information required in the result.
Due to its simplicity, people without a software modelling background can easily understand the model. Due to the specialised nature of the various cancer research sub-fields, often only the experts interviewed for defining each use case could provide us with the needed information for structuring the use cases and identifying the sources for each result. This has been one of the main challenges we have had to cope with in defining it.

5. Using Domain Modelling to Master the Interoperability Problem

Defining the domain model is an important activity in any analysis process. This is particularly true in the context of e-science projects where it plays a central role in the whole development process. A domain model represents a rigorously organized and selective abstraction of the knowledge in the domain experts' head. It addresses three main objectives: (1) acts as a bridge between the problem analysis and the solution design so it is used to ensure that the analysis that went into it applies to the final product, the software system; 2) is the *backbone* of the language used by all the team members, including analysts, domain experts and developers; (3) is the teams' agreed-upon way to structure the domain knowledge so that when new knowledge surfaces the domain model can be used to interpret it or to identify uncovered aspects. Defining the domain model is an iterative and incremental activity where rigorous analysis of meeting minutes, use cases, documentations, etc. by means of engineering techniques is intertwined with discussions with domain experts to reach a common vision. Moreover, the peculiarities of the e-science context require some properties to be considered in the model definition. Since the context is continuously evolving, it must be easy to accommodate inevitable changes. This requires the model to be extensible and flexible. Fine-grained and very specialized models are hardly extensible and it is often very difficult (if not impossible) to reach an agreement in heterogeneous teams involving domain experts with different specializations. In this light, our domain model has been defined to be sufficiently generic and to accommodate different points of view, while identifying the key entities and relationships. A second typical aspect of e-science projects is that their analysis requires a number of different perspectives to be considered [25]. As far as the perspective change, different typologies of entities and relationships may be required to describe the domain. For instance, in the above use case we can observe at least two perspectives. A *laboratory perspective* can be seen when the story talks about "specimens", "experiments", "protocol definition" (implicitly), "clustering services", etc. A *biological perspective* comes out in the last part of the use case's description when biological aspects are used to specify the searched information. On the other hand, since the platform will act as a system integrator allowing the interoperability among the different informatics resources, the *system integrator perspective* must be considered. The above mentioned data may be stored in "local repositories", "specialized bioinformatics repositories" or "Knowledge bases"; data can be represented in different "representation formats" so that "translation services" published on "service registries" may be needed to integrate them; "terminology sources" are used to annotate data and services so that these can be retrieved by the platform; and so forth. These aspects are not directly described in the use case but can be defined analysis from the system integrator perspective with the

multidisciplinary team. An important issue we have identified in this project is thus that in order to clearly identify all the needed information, a multi-perspective analysis is needed. This entails a multi-dimensional domain model to be defined for supporting the analysis activities. Each dimension can be considered a domain model by itself used alone or combined with the other dimension in the analysis activities.



Cancer Research	Investigation, Experiment, Protocol, Individual Information, Research Sample, Clinical Record, Experiment Result, Policy, etc.
Cancer Biology	Gene, Protein, Pathway, Agent, disease, drug, target, tissue, SNP, etc.
System Integrator	Bioinformatics Repository, Ontology, Registry, Service, Data Format, Access Policy, Protocol, etc.

Figure 4: The lti-dimensional domain model

In our project, The Platform domain model has been organized across three dimensions. The ***Cancer Research*** which includes all the concepts involved in the investigation/experiment execution, like *samples*, *patient data, protocols*, *publishable data* (results of the experiment executions), *etc.* The ***Cancer Biology*** including concepts like *tumour*, *drug*, *gene*, *pathway*, *etc.*
The ***System Integrator*** whose aim is to model the environment where the platform will operate, the different types of available resources and their relationships. Lack of space prevents us from showing the three models but
Figure 4 shows the three dimensions of our domain model above and some examples of the included concepts below. The complete specification can be found on [19].

5.1 Examples use case analysis through the domain model

In this section we show some excerpts from the analysis of the use case shown in Figure2. The GQR analysis is usually conducted together with the interviewed domain experts. Generally, the expert writes down the use case description and then, together with the analysis team, it is progressively structured whereby the method's concepts. The original description does not contain all the information that makes up the requirements. The GQR method has shown up to be effective in supporting the conversation between the analysts and the domain experts. For each question, experts are asked to identify what data sets and/or services may be used to produce the needed result, dragging in their knowledge of the context. Requirements are identified through this interaction and recorded as notes, at the beginning, and through semi-formal diagrams and tables afterwards.

Let us consider, as example, two questions among the five that follow from the goal.

- Q1.1 can be answered by identifying suitable specimens to be used in the experiment. This information can be found in tissue banks belonging to hospitals (e.g. Dundee tissue bank) or by means of global specialized repositories like those under construction in UK (e.g. OnCore [26]) or in US (e.g. caTissue [5]). The platform should thus be able to access to such repositories to query them and to produce the result R1, that is, a *list of specimens*. Using the cancer research perspective in the domain model, this result can be exalained by way of the entities *CR_ResearchSamples* and *CR_MedicalRecords*. The platform will thus need to search all repositories which are known to include data about such kind of entities. From the *system integrator* perspective, these repositories are modelled as *SI_InformaticsRepositories* (and in particular *SI_GlobalSpecializedRepositories* for "OnCore" and SI_LocaRepositories for "Dundee") and their data elements, representing the above mentioned entities, should be *semantically annotated* by metadata elements (*SI_MetaDataElements*) whose domain is defined within a *IS_TerminologySource* like the "NCI-Thesaurus" providing, for instance, concepts like *Organism, Disease, Disorder and Findings*, *Drugs and Chemical* to match the researcher's query attributes.
- Q1.4 can be answered by using the list of genes identified by the clustering service used to identify common genetic variations querying existing *knowledge bases* like REACTOME [6], PharmGKB [7], KEGG (Kyoto Encyclopedia of Genes and Genomes) [27], etc., or *scientific libraries* (e.g. PubMed) looking for papers which have reported about similar experiments. Currently, researchers use screen-scraping technologies to analyze and integrate data sets collected from different repositories.

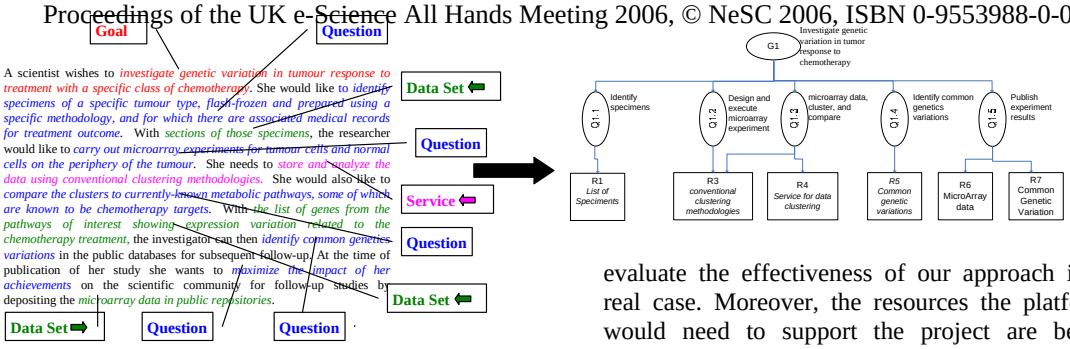


Figure 5: Excerpts from the use case analysis

R5 is the result of such integration that should be supported by the platform (as much as possible). Using the biological perspective in the domain model, we can explain the result in that saying that the platform should query repositories known to contain information about *CB_Gene*, *CB_Pathways*, *CB_ExpressionVariation* and *CB_Agent* (where a class of chemotherapy is considered an agent). From the *system integrator* perspective, these repositories are modelled as *SI_KnowledgeBases* (e.g. "REACTOME" and "PharmGKB") and *SI_ScientificLibrary* (e.g. PubMed). As well as in the previous example, their information should be annotated with proper metadata. To build the required result, the platform should integrate data coming from these sources by exploiting the semantic relationships among concepts defined in the *IS_TerminologySource* used as metadata source.

By these examples we have shown how the same use case is analyzed from different perspectives providing an explanation from both the user and system perspective on how the different resources should interoperate with one another to fulfil the high-level research goals.

6. Validation

The approach and models introduced in this paper are currently being validated in a companion demonstrator project (see [1] section Demo Projects, Imaging and Pathology). This project aims at developing services for 3D reconstruction of the macroscopic resected rectums using in vivo photographs and incorporating links to virtual histology images and MRI images. The project leverages technologies and services developed in the e.Diamond project [29] and is using data sets stored in repositories belonging to hospitals involved in a previous clinical trial, that is, the Mercury Trial [28]. Data (including images) is currently being collected and anonymized manually and transferred by coping it on CDs. Finally, the developed service has to be deployed so that it can be used in a distributed environment for more informed clinical decision making. Several user stories have been identified and are being analyzed in the context of this project to evaluate the effectiveness of our approach in a real case. Moreover, the resources the platform would need to support the project are being collected. These range across the different kinds of resource identified in section 2. Analyzing how they interoperate with one another to fulfil the various goals identified in the user stories will provide a suitable test-bed for validation of the whole information architecture.

7. Discussion and Future Work

In this paper we have recounted our experience in the early analysis of the NCRI Platform, a software system whose main aim is to enable the creation of a community supporting initiatives operating in the cancer research field in UK and world-wide. A multidisciplinary team composed by requirements engineers and domain experts was set up to carry out the analysis activities. Preliminary steps in the analysis process have been the identification of a core set of stakeholders and the main entities operating in the cancer research context, and the definition of a preliminary domain model to gain an initial common understanding of the problem within the multidisciplinary analysis team. Afterwards, the analysis has been driven by a set of use cases representing examples of research investigations that might benefit from the platform introduction. The main aim of our analysis has been to clearly understand how the various high heterogeneous resources making up the cancer research context might interoperate with one another to fulfil researchers' goals. The high heterogeneity of the cancer research domain and the need of working in a multidisciplinary team have required us to develop a customised approach to analyze the acquired use cases. The approach involves using an ad-hoc defined high level investigation model and a domain model. The former allows the analysis team along with the interviewed domain experts structuring an investigation throughout its lifecycle pointing out what researchers would ask the platform with and what resources it can answer with. The latter, allows analyzing in details the multiple facets of the interoperability problem by facing them from three orthogonal dimensions representing the researcher, biology and system points of view. The approach has proven to be effective in analysing a variety of use cases taken from different heterogeneous sub-

domains of the cancer research and particularly suitable for working with multidisciplinary teams.

Although it has been defined and used in the context of cancer research, we believe that the basic ideas and principles can be of interest to other projects in the broader e-research field.

Besides supporting the analysis activities, the investigation and domain models, mapped out in this paper, constitute the key information model around which the platform will be built. To achieve the platform's objectives, a semantic web service architecture is being designed following the Semantic Web Service Initiative proposed framework [30]. The platform will offer an environment where services requests (issued by the platform users) and offered services (issued by *service providers*) can be published. Adopting a model similar to the introduced Investigation Model, the platform will permit a simple workflow-like description of investigations where questions will correspond to service requests while results to offered services or to a set of available services integrated by *matchmaker services* provided by the platform environment. These services will be made accessible by providing semantic descriptions of their capabilities whose core concepts root in the Domain Model entities. These provide a high level conceptualization of the platform domain from three fundamental perspectives. This conceptualization can be used as a high-level ontology for semantically describing the web services offered by our platform. Such high level conceptualization will then be combined with a number of specialized ontologies (most of which already exist in the field) through ontology composition as described in [31].

Finally, we are currently analyzing and working in cooperation with several organizations which have already developed some of the informatics components the platform will need to use. Among others, we can mention the NCICB in US; the EBI; CancerGrid, CLEF and others.

Acknowledgements

The project team wish to acknowledge the funding for this work provided by Cancer Research UK and to thank all the researchers who have cooperated in the analysis activities.

References

[1] NCRI Informatics Initiative www.cancerinformatics.org.uk
[2] Standish Group, CHAOS report, www.standishgroup.com/chaos.htm
[3] Alistair Cockburn: Writing Effective Use Cases. Addison Wesley, 2001.
[4] European Bioinformatics Institute, http://www.ebi.ac.uk
[5] National Cancer Institute – Center for Bioinformatics, http://ncicb.nci.nih.gov/
[6] http://www.reactome.org/
[7] http://www.pharmgkb.org/
[8] http://www.ebi.ac.uk/swissprot/
[9] http://www.plos.org/
[10] http://www.ncbi.nlm.nih.gov
[11] http://www.geneontology.org/
[12] http://www.dh.gov.uk/PolicyAndGuidance
[13] www.hl7.org.uk/
[14] http://www.cdisc.org/
[15] http://www.mged.org/
[16] R. Begent, et al.: Challenges of Ultra Large Scale Integration of Biomedical Computing Systems", 18th IEEE International Symposium on Computer-Based Medical Systems, Dublin, Ireland, 2005.
[17] http://www.cancergrid.org/
[18] http://www.clinical-escience.org/
[19] www.cs.ucl.ac.uk/CancerInformatics
[20] I. Alexander, S. Robertson, S.: Understanding project sociology by modelling stakeholders. IEEE Software, vol. 21(1), Jan. 2004.
[21] Kuniavsky, M.: Observing the user experience: A Practitioner's Guide to User Research. Morgan Kaufmann, 2003
[22] M. Cohn: User Stories Applied: For Agile Software Development. Addison-Wesley, 2004
[23] A. Dardenne, A. van Lamsweerde, S. Fickas: Goal-Dircted Requirements Acquisition. Science of Computer Programming, 20 (1993)
[24] R. Solingen, E. Berghout: The Goal/ Question/ Metric Method McGraw-Hill, 1999
[25] A. Finkelstein, et al.: Viewpoints: a framework for integrating multiple perspectives in system development" Int. Journal of Software Engineering and Knowledge Engineering, vol. 2, 1992
[26] http://www.ntrac.org.uk/
[27] www.genome.jp/kegg/
[28] http://www.pelicancancer.org/researchprojects/mercury.html
[29] http://www.ediamond.ox.ac.uk/
[30] M. Burstein et al.: A Semantic Web Services Architecture. Internet Computing, Sept. 2005
[31] J. Jannink, P. Srinivasan, D. Verheijen G. Wiederhold: Encapsulation and composition of ontologies, Proc. of AAAI Summer Conference, AAAI (1998)

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Providing an Effective Data Infrastructure for the Simulation of Complex Materials

L. Roberts, L.J. Blanshard, K. Kleese van Dam

CCLRC eScience Centre, Daresbury Laboratory, Warrington, WA4 4AD

S. L. Price, L.S. Price and I. Brown

Department of Chemistry, University College London, 20 Gordon Street, London, WC1H 0AJ

Abstract

CCLRC have developed a suite of data management tools for the Engineering and Physical Sciences Research Council (EPSRC) funded e-Science project 'The Simulation of Complex Materials' [1], which ran from 2002 - 2005. The focus of the project was to aid the development of a computational technology for the prediction of the polymorphs of an organic molecule prior to its synthesis, which would then provide the ability "*to control the unwanted appearance of polymorphism and to exploit its benefits in the development, manufacture and processing of new molecular materials"* [2]. Prior to the project the data of interest was distributed across a multitude of sites and systems, with no simple formal methods for the management or distribution of data. This was considerably hindering the analysis of the results and the refinement of the prediction process and it was therefore essential to rationalise the data management process. The initial concern was for the collection and safe storage of the raw data files produced by the simulations during the computation workflow [3]. This data is now stored in a distributed file system, with tools provided for its access and sharing. As the data was not annotated with metadata it was difficult for it to be discovered and reused by others and so web interfaces were implemented to enable the cataloguing of data items and their subsequent browsing. In addition there was no fine grained access to the data. Specific crystal data is now parsed from the simulation outputs and stored in a relational database, and a web application has been deployed to enable extensive interrogation of the database content. This paper will elaborate on these tools and describe their impact on the achievement of the project's aims.

Background

A crystal may have different polymorphs, (different arrangements of the molecules in the crystal lattice), and

"different polymorphs have different physical properties, and so there are major problems in quality control in the manufacture of any polymorphic organic material. For example, a polymorphic transformation changes the melting point of cocoa butter and hence the taste, the detonation sensitivity of explosives producing industrial accidents, and the solubility changing the effective dose of pharmaceuticals." [2].

Therefore a method of predicting which crystal structure a given organic molecule will adopt under different conditions would have considerable benefit in product development across the range of molecular materials industries.

The computational chemistry group at UCL have developed computational methodologies for predicting the energetically feasible crystal structures of small, rigid, organic molecules [4]. Each simulation involves the running of multiple programs to generate these crystal structures, at considerable computational and human expense.

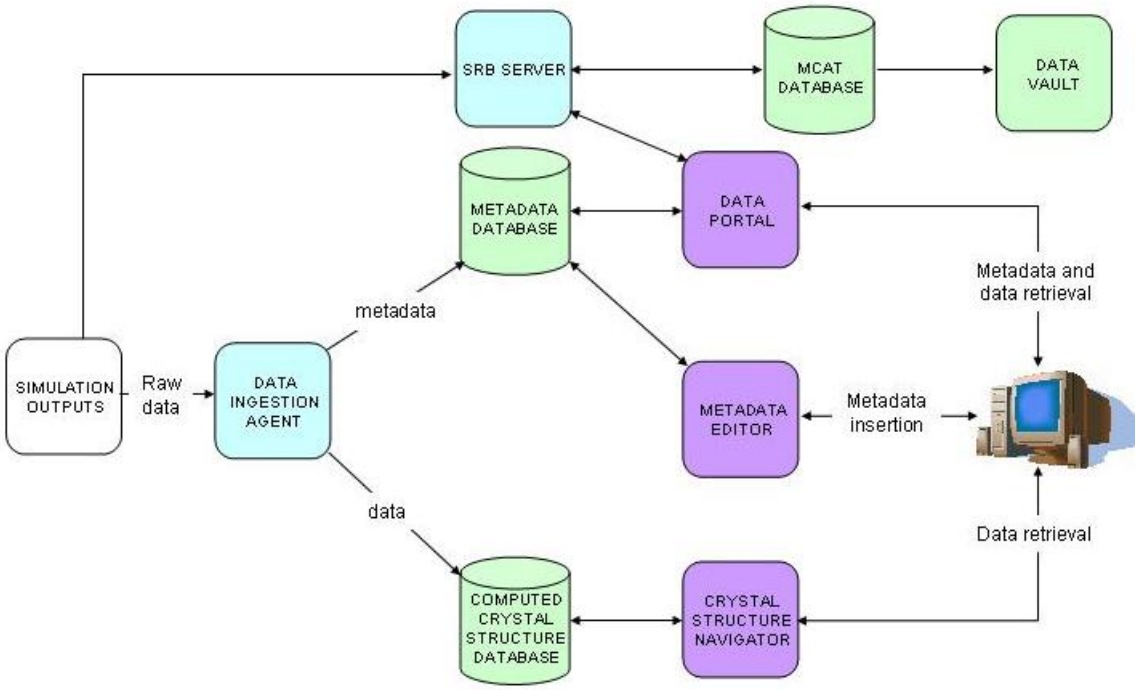


Figure 1: Deployed architecture for the eMaterials project

The process produces a great quantity of heterogeneous data estimating various mechanical, morphological and spectroscopic properties of the computed crystal structures. However there was little support for accessing and managing this data and this hindered the project's progress and collaborative efforts. To counter this CCLRC developed an effective data management infrastructure to facilitate the creation, storage and analysis of data and metadata.

Architecture

This infrastructure, as shown in figure 1, enables a more efficient cycle of discovery, analysis, creation and storage of both data and metadata. The tools provided for the management and discovery of metadata are the Data Portal [5], the Metadata Editor [6], the CCLRC metadata schema [7] and the metadata database [8]. The tools for the management of raw and processed data are the Computed Crystal Structures Database, the Crystal parser, the Crystal Structure Navigator and the Storage Resource Broker (SRB) [9]. The SRB was developed by the San Diego Super Computing Centre (SDSC).

After a simulation run the results are uploaded to the relevant storage media and can be discovered and shared with other scientists - for example, to use for comparison when a new polymorph is discovered experimentally that has already been computationally predicted. [10]

From the simulation outputs the raw data is processed in three ways:

i. A subset of the data is parsed from the files and inserted into the crystal database.
ii. The data files are uploaded into SRB.
iii. Metadata about the data is entered into the metadata database using the metadata editor.

Data Portal can then be used to discover and retrieve raw data from the SRB, and the Crystal Structure Navigator can be used to extract and compare data on the crystal structures from the computed crystal structure database.

Storage Resource Broker (SRB)

The first issue to be overcome was the lack of tools to manage the collection and safe storage of simulation outputs, which also made it difficult to share data. Data files from simulation runs were generally located on users' machines or on the machine on which the computation had been run.

In 2002 SRB was chosen as the middleware because of its appropriateness and CCLRC's strong links with SDSC. GUIs (Data Portal and Metadata Manager) that used the Scommands behind the scenes were written to allow users access to the grid enabled storage resources.

SRB is a distributed file system that allows files to be shared across multiple resources whilst providing access through a single interface and showing a unified logical view of the virtual organisation's data. As such data is organised in a logical rather than a physical context, and the complexity of locating and mediating access to individual data items is abstracted away from the users. Access control is simply implemented with the data originator able to control and configure access permissions to their data for other project members. The raw data produced by the simulations is now stored on a number of distributed resources managed by the SRB.

Metadata Database

Although the use of SRB facilitates data distribution and replication, the value of the data is in its discoverability – and the annotation of the data holdings with metadata is what achieves this.

The raw data files in SRB are now catalogued with metadata, and this data is stored in the metadata database which uses the CCLRC metadata schema. This is a common model that has been re-used in other CCLRC projects that required metadata holdings. As the same logical model has been re-used across different projects the applications written against this model, (Data Portal and Metadata Manager), have also been reused on other projects, with very little customisation required.

Raw data files are grouped into datasets, and datasets are grouped under studies with a name, start date, end date and originator details. The studies are also linked to topics, keywords and investigator details. The database tables for the data file and data set entities also store the physical location of the data in SRB, alongside the metadata.

Metadata Manager

The CCLRC metadata manager is a web-based tool for the insertion and manipulation of entries in the metadata database. Typically users annotate their datasets and data files with details such as the provenance of the data and its location in SRB. Users can organise their data by creating and editing information about studies and then adding datasets and data files to the hierarchy. The metadata forms the basis for the search and retrieval of data by the Data Portal.

In the most recent version of Metadata Manager users can also add topics, so removing the last vestiges of a centrally controlled vocabulary in the infrastructure. Scriptable command line tools for the insertion of metadata have also now been written.

Data Portal

The CCLRC Data Portal is a web-based front-end that enables the scientists to browse the metadata database and discover data resources from physically diverse institutions. Data is displayed in a virtual logical file system – so files are displayed by topic (such as the molecule) rather than by geographical location, this being of much more use to the project participants. The required data resources can then be downloaded from SRB directly to the users' machines.

The Computed Crystal Structure Database

Although SRB allowed the searching of related data it did not provide any tooling for data comparisons and refinement as there was no fine grained access to the data, which was still stored in unstructured text files. This hindered analysis of the results and it was very difficult to look at, for example, all the crystal structures with a total lattice energy beneath a certain threshold. Therefore the project requested a bespoke database (figure 2) specific to their requirements

and data, which would enable them to drill down and query across data.

The CCLRC Computed Crystal Structure Database is a relational database running on Oracle 10G and is hosted on the National Grid Service (NGS). It is used to store specific data about crystal structures, rather than metadata or unprocessed raw data in a file or CLOB format. The data model for the calculated crystal database was designed to allow easy interrogation of the database and provide good performance for complex queries as the database grew.

There are five tables for representing the crystal data –

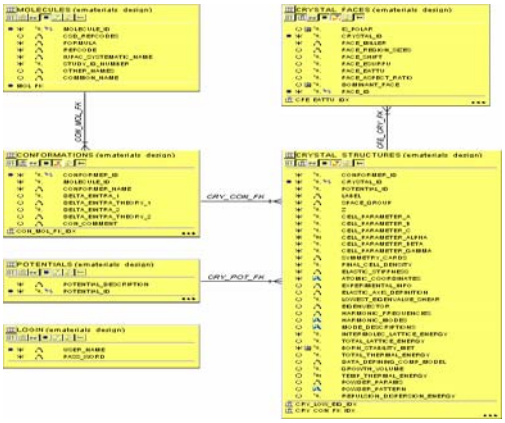


Figure 2: Crystal database schema

molecules, conformations, potentials, crystal_structures, and crystal_faces. At the moment it is quite a small database – holding just 42 molecules and around 2493 crystal structures with 228788 crystal faces between them. The attributes stored are molecule, conformation, potential and crystal descriptors, as well as growth volume attributes, second derivative properties and various energies. However, the database is being continually expanded as more molecules are studied, and the range of properties to be stored can be extended in the future if necessary. As more data has been added the database has scaled well and the applications have not suffered a drop in performance.

Populating the Database

An automated data ingestion agent, (written by Ian Brown at UCL), parses individual datum from the fixed format output files from the simulation runs and populates the computed crystal structure database with a subset of the simulation data. Some extra information about the molecule or conformation or potential has to be entered into a configuration file by the project administrator (Louise Price) and the script is then run from the command line. If any information on the study being uploaded is already contained in the database it is updated otherwise new records are inserted into the database.

Crystal Structure Navigator

The crystal structure navigator is a powerful query tool that provides users with the ability to search for crystal data which fits certain criteria by allowing the construction and execution of queries against the calculated crystal database, (figure 3). The application runs on a Tomcat web server and is written with JSPs and servlets. It has a simple and intuitive interface from which the user can compile their display and search options.

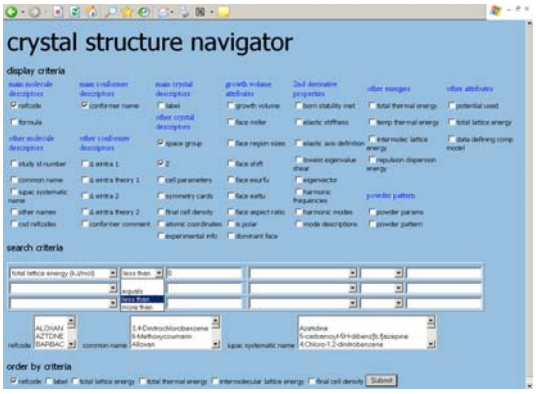


Figure 3: Crystal Navigator selection screen

From the display criteria section of the web page users can choose which combination of any of the forty-eight properties held in the database they would like to display. The search criteria section allows up to six criteria to be composed – with the user choosing the property, the relational operator and the value - such as 'refcode like E-N-I' or 'total lattice energy <-100'. To prevent user errors and simplify selection the relational operators are only enabled after the user has selected a property – for example if a user selects a textual property, such as common name, it would not make much sense to offer them 'more than' or 'less than' as operators; and likewise if they picked a numerical field such as total_lattice_energy then the

comparative operator 'like' would not be offered to them. There is also a section of 'quickpicks' – which shows a non-static list of values for the attributes refcode, common name and iupac_systematic_name for the user to choose from. This list is updated automatically as new entries are made in the database. Lastly the user can choose to impose an ordering on the results – for example the results can be ordered alphabetically by refcode, or by total_lattice_energy ascending.

Thus the complexity of relating the data fields, joining the relational tables and restricting the set of data to be returned is all hidden from the user whilst most of the functionality of SQL is made available in a user friendly interface. The results are then displayed on a results page, with only those data that fit the user's criteria and only those properties that the user selected being displayed. The user can download the results into a spreadsheet and this is now being used for the primary scientific analysis and for publications, and can also be used as the basis for input into other simulations. If the user wishes to perform a new search they can bring up a new criteria selection page, or they can review the criteria they just submitted.

Conclusion

The project has supplied an effective infrastructure for the management and utilisation of data. The provided suite of software is used daily to aid the computational studies on the polymorphism of various molecules being performed by the “Control and Prediction of the Organic Solid State” [2] project of the Research Council UK’s Basic Technology Program. The project uses the tools to make comparisons across the range of molecules being studied to refine and develop techniques for polymorph prediction. The feedback received from the scientists on the project has been extremely positive and the software suite is now essential for the storage, discovery and analysis of their data. The components of the architecture that deal with metadata have been reused on other projects with only a small and easy amount of customisation has been required.

References

[1] Simulation of Complex Materials project http://www.e-science.clrc.ac.uk/web/projects/complexmaterials

[2] Brief Overview, Control and Prediction of the Organic Solid State http://www.cposs.org.uk

[3] Blanshard, L.; Tyer, R.; Kleese van Dam, K. “eMaterials: Integrating Grid Computation and Data Management Services”; UK e-Science All Hands Meeting, 2004, Nottingham, UK.

[4] Price, S. L. "The Computational Prediction of Pharmaceutical Crystal Structures and Polymorphism." *Adv. Drug Deliver. Rev* **2004**, *56*, 301-319.

[5] Drinkwater, G.; Kleese van Dam, K.; Manandhar, A.; Sufi, S.; Blanshard, L. “Data Management with the CCLRC Data Portal”; International Conference on Parallel and Distributed Processing Techniques and Applications, 2004, USA.

[6] CCLRC Metadata Editor http://www.e-science.clrc.ac.uk/web/projects/scientific_metadatamgnt

[7] CCLRC Scientific Metadata schema http://www.escience.cclrc.ac.uk/documents/staff/shoaib_sufi/csmdm.version-2.doc

[8] Blanshard, L.; Kleese van Dam, K.; Catlow, C. R. A.; Price, S. L. “Simulation of Complex Materials: Database Design for Metadata”; UK e-Science All Hands Meeting, 2003, Nottingham, UK.

[9] SRB Home Page http://www.sdsc.edu/srb/index.php

[10] Vishweshwar, P.; McMahon, J. A.; Oliveira, M.; Peterson, M. L. & Zaworotko, M. J. "The Predictably Elusive Form II of Aspirin." *J. Am. Chem. Soc.* **2005,** *127*, 16802-16803

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Secured bulk file transfer over HTTP(S)

Yibiao Li and Andrew McNab

Manchester University

Abstract

A method of secured bulk file transferring over HTTP(S) for Gridsite is discussed in this article. Unlike FTP, this method can transfer a file from one Grid node to another Grid node directly, using zero memory of the client computer. The verified information is transferred over HTTPS while the file is transferred over HTTP. To speed up the file transfer, a multi-connection technique is adopted.

Keywords: bulk file, HTTP(s) transfer, GridSite, GridHTTP protocol

1. Background

GridSite[5] was originally a web application developed for managing and formatting the content of the GridPP[8] website. Over the past three years it has grown into a set of extensions to the Apache web server and a toolkit for Grid credentials, GACL access control lists and HTTP(S) protocol operations. A powerful client end command, htcp, was developed for user to operate (delete, move, copy etc) file/directory on GridSite nodes. Recently, a functionality of the bulk file transfer was added into the command, which can now transfer a bulk file between two GridSite nodes directly without using the memory of the local machine.
For the ease of the reader's understanding, here we briefly introduce the GridSite.

1.1GridSite node

Each GridSite node is equipped with apache[6] and GridSite package. Normally, a GridSite node can be accessible by both HTTP and HTTPS. The authorized user (see the following section) can “write” or “update” the contents of GridSite node over HTTPS besides reading the contents of it over HTTP.

1.1GridSite authentication and authorization

To access GridSite node over HTTPS, user must have a user certificate issued by related Certification Authority (CA). A user certificate usually has a version of user's name and affiliation as its Distinguished Name (DN) - for example,
"/C=UK/O=eScience/OU=UniversityName/L=GroupName/CN=FirstName Surname".
Once the user has obtained a user certificate in his name from his CA, the user needs to make sure it is loaded into the browser the user normally uses to browse the web. Browsers want the certificate and private key in the PKCS#12 format, which is normally a single file with the extension ".p12". Many programs which are based on OpenSSL, such as Globus and curl, prefer the PEM (".pem") format for certificates, with separate certificate and key files ("usercert.pem" and "userkey.pem"). These two formats can be easily converted to each other with software tools.
Once the user certificate is loaded into the browser, the user should be able to see his/her certificate name appear when looking at an HTTPS GridSite page which has the page footers enabled. If GridSite understands the user certificate, it displays a "You are ..." line in the footer.
Once users access a GridSite node with their identity, they will be authorized appropriate rights depending on their identity. GridSite allows site administrators to specify these rights for individuals and groups using GACL access control files (see next section). GACL defines who can read files, who can list directories, who can write or create files and who can modify the GACL policy files. To get increased access to an area of a site, the user needs to contact the administrator for that area and give the DN of the user担 certificate (it's not necessary to send any certificate files.)

1.3 Access Control

DN Lists appear in the Grid Access Control Lists (GACL) used by GridSite. These are stored as .gacl files in directories: if the .gacl file is present, it governs access to the directory; if it is absent, then the parent directories are searched upwards until a .gacl is found.
The GridSite GACL Reference explains the XML format of these files, but they can be

edited using the ACL editor built into the GridSite system by people who have the Admin permission within the ACL.
If a user has this permission in a given directory, when the user views directory listings or files with a browser in that directory the user will see the option "Manage Directory" in the page footer. This allows the user to get a listing of the directory and the .gacl file will appear at the top if it's present. If not, then there will be a button to create a new .gacl file with the same permissions as have been inherited by that directory from its parent.
GACL allows quite complex conditions to be imposed on access, but normally user can think of an ACL as being composed of a number of entries, each of which contains one condition (the required credential) and a set of allowed and denied permissions.
Credentials can be individual user's certificate names or whole groups of certificate names if a DN List is given. (User can also specifiy hostname patterns using Unix shell wildcards (eg *.ac.uk) or EDG VOMS attribute certificates - see the GACL Reference for details.)
Permissions can be Admin (edit the ACL), Write (create, modify or delete files), List (browse the directory) or Read (read files.) Permissions can be allowed or denied. If denied by any entry, the permission is not available to that user or DN List (depending on what credential type was associated with the Deny.)

2. Why transfer files over HTTP(S)?

Normally, there are the following way to transfer files over internet.

2.1 Email

One of the most important aspects of the Internet is the ability to send large files easily. Email is still the primary way to receive or send large files over the Internet. Unfortunately, using Email to send large files or receive large files is fraught with drawbacks. In today's world, file sizes are getting larger and larger but email technology has not advanced at the same pace. It is no longer efficient to send large files via Email and in many cases it is impossible.

2.2 FTP

FTP is a method for exchanging files over the internet utilizing standard TCP/IP protocols to enable data transfer.
FTP can be used to upload and download files of almost any size from or to a central server. It is a well-established and consistently implemented protocol that can be enabled on the Windows Storage Server.
The advantages of FTP include:
- Support for all kinds of clients: Standardized implementation of the protocol means that virtually any FTP client, running on a Microsoft or non-Microsoft operating system, can use the FTP server.
- High performance and simplicity: Performance and simplicity of the protocol makes it a convenient option for file transfers across the Internet.

The primary disadvantage of FTP is that data and logon information is sent unencrypted across the network. This could result in the discovery of logon accounts or passwords. This information could be used by unauthorized individuals to access other systems.

2.3 HTTP

The HTTP[1] protocol is a protocol for file transfer over internet. It is often used to download HTML files or image files through a web browser such as IE, Mozilla, or FireFox. But it can also be used to file upload by some command under Unix/Linux OS.

2.4 HTTPS

HTTPS is a communications protocol designed to transfer encrypted information between computers over the Internet. HTTPS is HTTP using a Secure Socket Layer (SSL).
It is recommended that users utilize HTTPS when transferring files containing security sensitive information.

3. Design

3.1 GridHTTP protocol

To realize the file transfer between GridSite nodes, GridHTTP protocol was designed, which supports bulk data transfers via unencrypted HTTP while the information of authentication and authorization with the usual grid credentials over HTTPS.
To initiate a GridHTTP transfer, clients set an `Upgrade: GridHTTP/1.0` header when making an HTTPS request for a file. This header notifies the server that the client would prefer to retrieve the file by HTTP rather than HTTPS, if possible. The authentication and authorization are done via HTTPS (X.509, VOMS, GACL etc deciding whether it is right) and then the server may redirect the client to an HTTP version of the file using a standard HTTP 302 redirect

response giving the HTTP URL (which can be on a different server, in the general case.) For small files, the server can choose to return the file over HTTPS as the response body. When contacting a legacy server, the Upgrade header will be silently ignored and the file will be returned via HTTPS as normal.
For redirection to plain HTTP transport, a standard HTTP Set-Cookie header is used to send the client a one-time pass-code in the form of a cookie, GRIDHTTP_PASSCODE, which much be presented to obtain the file via HTTP. This one-time pass-code only works for the file in question, and only works once: the current implementation stores it in a file and deletes the file when the pass-code is used. (This mechanism is no worse than GridFTP for providing an unencrypted data channel: it's vulnerable to man-in-the-middle attacks or snooping to obtain a copy of the requested file, but not vulnerable to replay attacks or to other files being obtained by the attacker.)
As you can see, GridHTTP is really a profile for using the HTTP/1.1 standard, rather than a new protocol or a set of extensions: no new headers or methods are involved.
Ways of extending it to support variable TCP window sizes so it can be used for a mix of long and short distance connections (currently the TCP window size has to be set in the Apache configuration file), and support for third-party transfers using the HTTP COPY method from WebDAV are being added to the GridSite implementation.

3.2 Advantages

One big advantage of redirecting to a pure HTTP GET transfer is not just that the server and client don't have to spend CPU en/decrypting it, but that Apache can use the sendfile() system call to tell the kernel to copy it directly from the file system to the network socket (or can use the Linux kernel module HTTP server, which has much the same effect.) This means the data never has to be copied through user space (the so-called zero copy mode.)
As far as client side APIs go, any client side library which supports HTTP redirects and cookies and lets user add his/her own headers is sufficient (even the curl command line tool lets user do this, with the -H and -c options, without having to make any modifications to its code.)
From GridSite version 1.1.11, htcp supports GridHTTP redirection, by using the --grid-http option.

3.3 Bulk file transfer between GridSite nodes

Assume that a grid user wants to copy a bulk file from GridSite node (source server) to another GridSite node by giving a batch of commands (so he cannot logon destination to copy file directly) on a computer denoted as client computer.
Now we consider a simple case, copying files without secured factor.
In general, using a command like wget or curl, the user can copy the file to the local computer first, then upload it to the destination server as shown in figure 1.

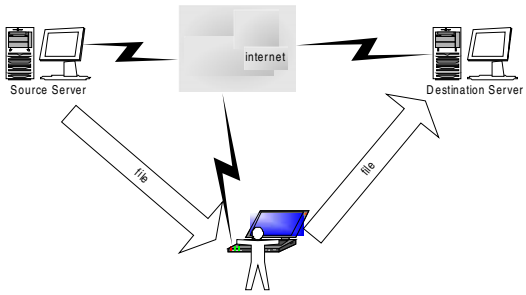


Fig 1 user downloads file first to local computer, then uploads it to destination server

Though this method can do the job, it apparently waste time, internet bandwidth and local machine memory and disk space.
Instead, we can seek a way to send command to destination server, and ask it to get file from source server.

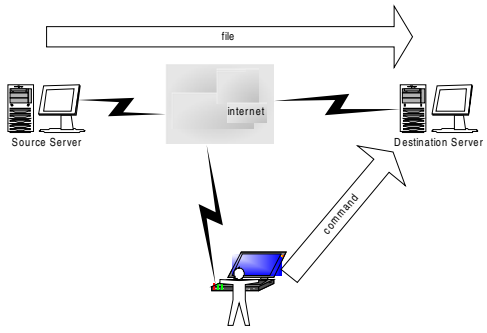


Fig 2 user sends command to destination server to ask it to copy file from source server

To realize it, there should be a module on the server side, which can:

1. receive a http request,
2. retrieve the file information from the request package,
3. send request to source server to get the file,
4. receive file and save it.

According to extendible feature of the Apache, we can develop such a module and attach it to the Apache.

Now let us add the secured feature to the above case in GridSite circumstance.

To support the remote bulk file copy, the GridSite node should:

1. verify the user certificate (source server).
2. produce one-tine pass-code (source server).
3. check the user access to the destination directory, respond to the user if no write access (destination server).
4. send file request with pass-code as a cookie via HTTP (destination server).
5. redirect HTTPS request to HTTP (source server).
6. retrieve the file and save it to some directory (destination server).
7. respond to user when finishing (destination server).

The client end command should do the following:

1. send a file pass-code request to source GridSite node with the user certificate over HTTPS
2. retrieve pass-code from source GridSite node over HTTPS
3. send file copy request to destination GridSite node with the pass-code over HTTPS

A completed description of the bulk file copy system can be given as (see figure 3):

1. Client uses user ID (user certificate and user key) to request a one time pass-code from source gridsite server (HTTPS)
2. Source server verifies the user ID.
3. Source server issues a onetime pass-code to client (HTTPS)
4. Client sends a request to destination server to get file from source server with onetime pass-code (HTTPS)
5. Destination server sends request to source server with pass-code (HTTPS)
6. Source server verifies the pass-code
7. Source server transfer file to destination server (HTTP)

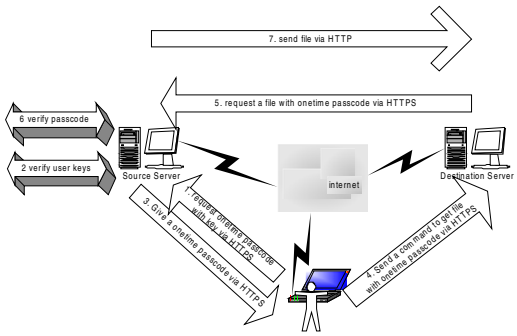


Fig 3 a completed bulk file transfer system (single connection)

Furthermore, considering that transferring a bulk file could take a lot of time, we can use the multi-connection technique to get different part of each in each connection at the same time. that will speed up the bulk file copying. But one problem is that when the destination server sends the request for file copying, a one-time pass-code is required for each connection, to get one-time pass-codes for connections (one pass-code for each connection), a secured connection have to be created to transfer the request and response of one-time pass-code, and the source server will have a mechanism to produce a one-time pass-code when it receives a request with original pass-code.

Thus the completed procedure for multi-connection bulk file copy can be described as:

1. User send a request to source server with uses user ID (user certificate and user key) to request a reusable time pass-code (HTTPS),
2. Source server verifies the user ID,
3. Source server issues a reusable pass-code to client computer if ID is OK, or responds a error message (HTTPS),
4. Client sends a request to destination server to get file from source server with reusable pass-code (HTTPS),
5. Destination server sends request to source server with reusable pass-code to get file size (HTTPS),
6. Destination server creates multi-connection, and get one-time pass-code for each connection (HTTPS),
7. Source server verifies the pass-code, and produce one-time pass-code and send it back to the destination server (HTTPS).
8. Destination server requests part of file in each connection with one-time pass-code (HTTPS),
9. Source server checks the one-time pass-code and transfer part of file to destination server (HTTP)
10. Destination server combines parts of file into a completed file, and save it to the directory required.

Please note that in step 9 and 10, if we change the HTTP connection to HTTPS connection, then the whole file will be transferred over HTTPS. So it is easy to extended to the case of transferring both file and security data over HTTPS.

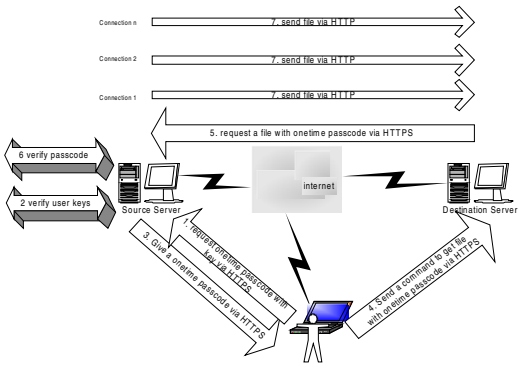


Fig 4 a completed bulk file transfer system (multi-connection)

4 Implementation

To realize the design in section 3, we consider the implementation on the server side and client side separately.

1.1server side

On the server side, we need to realize two main modules. The one is responsible for producing the one-time pass-code and reusable pass-code and relative check (source server side in the above discussion), the other is responsible for responding client requests and copying file (destination server in the above discussion).

In practise, we developed first module (pass-code) as part of GridSite, and developed the second module as a CGI program.

The pass-code module was part of GridSite package which can be compiled and run as an apache module[2][3]. the key task here is the one-time pass-code encoding, as described in the protocol, the pass-code should be related with one file including its full path and the time. Here we generate a random number and save it to some directory that is only accessible by the Apache server, once the file related to this pass-code has been sent, the pass-code will then be deleted, or after a specified period, even the file has not been sent, it will be deleted. This pass-code mechanism here ensures that even someone knows a pass-code by some means, he cannot get access right to other files.

The second module is responsible for receiving the user request and obtaining the file from source server. In the current version of GridSite package, we developed it as a CGI program, in Apache configuration file, it is mapped to HTTP COPY method. It was developed with C and libcurl, the key matter in the one connection case is to transfer the pass-code as a cookie via HTTPS connection, then get the file over HTTP. In the multi-connection case, we used the multi-thread technique. For each connection, we use the reusable pass-code to get the one-time pass-code over HTTPS, then to get the part of the file over HTTP.

4.2 Client side

On the client side, the command was built in the powerful command htcp. The options needed to pass to the destination server, such as connection number, block size of each connection and thread number specified by user are transferred with OPTS in the HTTP header over HTTPS.

5. The server configuration and command usage

5.1 The server configuration

As described in section 4.1, there are two modules in server side. The pass-code module has built in the GridSite package, so after the package is installed and the apache service starts, this module has been started, there is no extra configuration needed.

The second module is also included in the GridSite package but run as a separate program, named gridsite-copy.cgi, it should be installed into a specified directory that is normally mapped to /cgi-bin/ in apache configuration file. So after the GridSite package is installed, check the directory if you know where it is, or check the apache configuration file https.conf first to find out where it is, then check if the file gridsite-copy.cgi is there. The following line is needed to add in the apache configuration file httpd.conf:

Script COPY /cgi-bin/gridsite-copy.cgi

If you want to use copy a file from the GridSite node to another, you must have write access for the destination directory on the destination node. The access configuration is described in section 1.3 access control.

5.2 Client command

To copy a file from one GridSite node to another, you use htcp command provided by GridSite package. Here we give two examples.

Example 1: copy a file data.dat from node A to node B's data directory using one connection:

htcp rmtcp https://a/data.dat https://b/data/

Example 2: copy a file data.dat from node A to node B's data directory using multi-connection:

htcp -rmtcp -connection-number 5 -block-size 20 https://a/data.dat https://b/data/

Note that in the above examples, the option rmtcp ask htcp to execute the remote copy module; option connection-number indicates how name connections will be used to get the

file, and option block-size indicates the maximum k-byte for each connection.
To use htcp command, users should copy their certificates in a directory .globus or current directory.

6. Notes

There is another application tool for the GridSites to transfer files, GridFTP[9], which is based on the popular File Transportation Protocol FTP, supporting functionalities such as:
1) Grid Security Infrastructure (GSI)
2) Third-party control of data transfer
3) Parallel data transfer
4) Striped data transfer
5) Partial file transfer

As shown in the previous sections, the GridHTTP provides similar functionalities, but most difference from GridFTP is GridHTTP is based on HTTP protocol.
There are a lot of arguments on advantages and disadvantages of file transferring over HTTP or FTP. It is hard to determine which is better from the theory. For the grid environments, GridFTP needs extra installation and configurations, while the htcp can embedded into GridSite package as an Apache module and an independent CGI application program, and needs quite simple configurations.

The remote copy method for GridSite nodes discussed in this article can be easily applied to and developed in normal Apache nodes.

7Acknowledgements

This work was funded by the Particle Physics and Astronomy Research Council through the GridPP programme.

References:

[1] R. Fielding etc, Hypertext Transfer Protocol – HTTP/1.1", http://www.w3.org/ Protocols/rfc2616/rfc2616.html, 1999
[2] L. Stein and D. MacEachern, Writing Apache Modules with Perl and C", O'Reilly & Associates, 1999
[3] B. Laurie and P. Laurie
Apache: The Definitive Guide, Third Edition", O'Reilly & Associates, Third Edition, 2002
[4] Thomas Boutell, Featuring C and Perl 5 Source Code", Addison Wesley, 1996
[5] GridSite software and documents: http://www.gridsite.org
[6] Apache official website: http://www.apache.org
[7] Curl and libcurl stuff: http://curl.haxx.se
[8] GridPP website: http://www.gridpp.ac.uk
[9] GridFTP website: http://www.globus.org/toolkit/docs/3.2/gridftp/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Dynamic Data Replication in LCG 2008

C. Nicholson[1], D. G. Cameron[2], A. T. Doyle[1], A. P. Millar[1], K. Stockinger[3]

[1] University of Glasgow, Glasgow, G12 8QQ, Scotland

[2] CERN, European Organization for Nuclear Research, 1211 Geneva, Switzerland

[3] Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Abstract

To provide performant access to data from high energy physics experiments such as the Large Hadron Collider (LHC), controlled replication of files among grid sites is required. Dynamic, automated replication in response to jobs may also be useful, and has been investigated using the grid simulator OptorSim. In this paper, results from simulation of the LHC Computing Grid in 2008, in a physics analysis scenario, are presented. These show, first, that dynamic replication does give improved job throughput; second, that for this complex grid system, simple replication strategies such as LRU and LFU are as effective as more advanced economic models; third, that grid site policies which allow maximum resource sharing are more effective; and lastly, that dynamic replication is particularly effective when data access patterns include some files being accessed more often than others, such as with a Zipf-like distribution.

1 Introduction

The latest high energy particle accelerators, such as the Large Hadron Collider (LHC) at CERN, the European Organization for Nuclear Research, are eagerly awaited by particle physicists. The LHC alone is expected to produce tens of petabytes of raw data annually, making it necessary to distribute computing and storage resources in a worldwide grid. Specific to the requirements of the LHC is the LHC Computing Grid (LCG), which uses a three-tiered grid architecture. In this architecture, raw data is produced at the CERN central Tier-0; it is replicated in a controlled fashion to a number of Tier-1 sites, which are responsible for permanent storage. Each Tier-1 has a number of associated Tier-2 sites, providing computing power for user analysis along with modest storage capabilities.

In such a data-intensive grid, replication of files among grid sites is important to improve grid performance. Apart from the kind of controlled replication planned for LCG, however, dynamic data replication - automatically replicating files between sites in response to jobs - may also be useful in achieving an optimal distribution of data around the grid, and there are numerous possible strategies for such dynamic replication. Current grids are not yet mature enough to allow testing of this kind of replication, however, so the grid simulator OptorSim [1] has been designed to explore the effects of dynamic data replication under a variety of conditions.

In previous work, OptorSim was used to study various grids including the European DataGrid [3] and the LCG topology of 2004 [6]. In this paper, the much more complex setting of LCG in 2008 (the first full year in which the LHC will produce data) is investigated, evaluating some simple replication strategies as well as an economic model of file replication, under a range of conditions. First, a brief survey of related work is given. The grid optimisation principles and replication strategies investigated are then presented, followed by a short description of OptorSim in Section 4. The experiments performed are described in Section 5 and the results presented in Section 6, before drawing some conclusions in Section 7.

2 Related Work

In recent years there have been several grid simulation projects, examining various aspects of grid systems. The *Models of Networked Analysis at Regional Centres for LHC Experiments*, or MONARC, project was initiated to explore computing models for the LHC projects. As part of this project, a simulator was developed to provide a realistic simulation of distributed computing systems with which different data processing architectures could be evaluated [10]. Its main difference from OptorSim, however, lies in the lack of infrastructure for automated replication and replica optimisation.

GridSim [5] is a grid simulation toolkit developed to investigate resource allocation techniques and in particular, a computational economy. The focus is on scheduling and resource brokering; there is not, however, capability for data management such as would be required for investigation of replica optimisation strategies. ChicSim [12] is a simulator designed to investigate scheduling strategies in conjuction with data location. Bricks Grid [13], while initially focusing on job scheduling, has been extended to include replica management. Its replica management components, however, are centralised rather than the distributed architecture used in OptorSim. All these projects therefore give a complementary approach to that used in OptorSim, allowing exploration of different areas of parameter space.

3 Grid Replica Optimisation

In a grid environment, there are many variables which determine its overall performance, and which are impossible to harmonise into one optimal configuration for the whole grid. It is possible, however, to optimise those variables which are a part of the grid middleware itself. For a data grid, the most important areas to optimise will then be job scheduling and data management. This paper will concentrate on data management, and dynamic replica optimisation in particular.

3.1 A Replica Optimisation Service

There are several important design decisions behind the OptorSim replication model. First, it is distributed rather than centralised or hierarchical. Each site is able to manage its own replica content and there is no single point of failure; it also removes the need for sites to know the state of the whole grid.

Second, it operates on a pull rather than a push model. In a push model, the site containing a particular data file would decide when to replicate it and where to. In a pull model, a site which did not initially have the file would decide when to replicate it to itself, and where from. In a real particle physics grid, however, sites would be unlikely to accept spontaneous replication of data from some other site and so a pull model is favoured, with each site responsible for its own replica optimisation.

Finally, a replication trigger must be chosen. When a site is requested for a file which it does not have, for example, this could trigger the first stage of the replication strategy. Another possible trigger could be a file on some other site reaching a certain level of popularity. This would require monitoring of all file popularities, however. The simplest approach is to trigger on a file request, and this is what has been implemented in OptorSim.

The overall aim of such a distributed replication model would be to achieve global optimisation as a result of local optimisation by each site's Replica Optimiser (RO). Each RO therefore has two goals: the minimisation of individual job execution costs, and the maximisation of usefulness of locally stored files. A good replication strategy will be one which achieves a good trade-off between individual running times and overall resource utilisation.

3.2 Stages of a Replication Strategy

Replication can be logically separated into three stages through which any replication strategy must proceed. These are delineated as follows, with each stage depending on the success of the preceding stage. First comes the *Replication Decision* where, given the trigger condition, the RO at a site must decide whether or not to replicate the file to its local site. If it decides not to replicate, the file must be read remotely. The second stage is *Replica Selection*, where if the RO has decided to replicate the file, it must choose which existing replica to copy. Finally, in the *File Replacement* stage, if the local site does not have sufficient space to store the new replica, one or more files must be deleted until there is enough space. These three stages will each be discussed for the replication strategies which are described in Section 4.

4 OptorSim

OptorSim is an event-driven simulator, written in Java. As dynamic data replication involves automated decisions about replica placement and deletion, the emphasis is on simulation of the replica management infrastructure. The architecture and implementation are described in [8] and so only a brief description is given here.

4.1 Architecture

The conceptual model of the OptorSim architecture is shown in Figure 1. In this model, the grid consists of a number of sites, connected by network links. A grid site may have a Computing Element (CE), a Storage Element (SE) or both. Each site also has a Replica Optimiser (RO) which makes decisions on replications to that site. A Resource Broker (RB) handles the scheduling of jobs to sites, where they run on the CEs. Jobs process files, which are stored in the SEs and can be replicated between sites according to the decisions made by the RO. A Replica

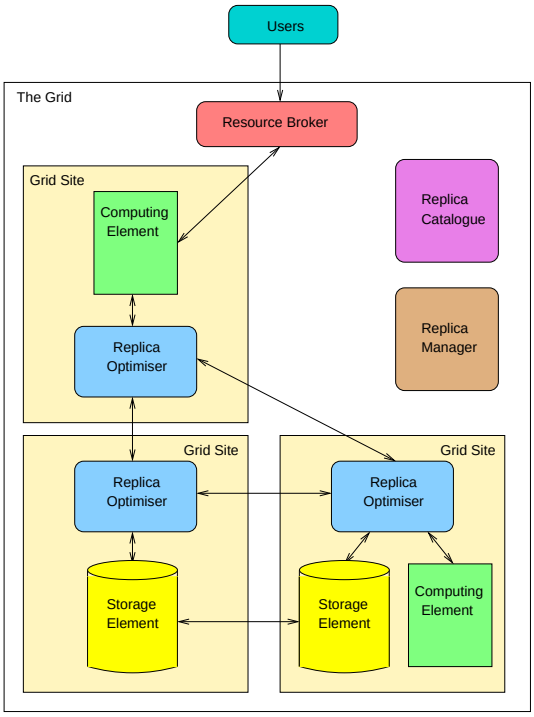


Figure 1: Grid architecture used in OptorSim.

Catalogue holds mappings of logical filenames to physical filenames and a Replica Manager handles replications and registers them in the Catalogue.

4.2 Simulation Inputs

4.2.1 Grid Topology.

To input a grid topology, a user specifies the storage capacity and computing power at each site, and the capacity and layout of the network links between each. SEs are defined to have a certain capacity, in MB, and CEs to have a certain number of "worker nodes" with a given processing power. Sites which have neither a CE nor an SE act as routers on the network. Background traffic on the network can also be simulated.

4.2.2 Jobs and Files.

A physics analysis job usually processes a number of files. This is simulated by defining a list of jobs and the files that they need; a job will process some or all of the files in its dataset, according to the *access pattern* which has been chosen. The time a file takes to process depends on its size and on the number and processing power of worker nodes at the CE.

4.2.3 Access Patterns.

Several file access patterns have been implemented in OptorSim, aiming to simulate various possible grid scenarios. These include *sequential* access, where a job accesses each file it requires once, in sequence; and *Zipf* [1], where a few files are accessed many times while others are accessed infrequently. It has been shown in [9] that both cases can occur in a particle physics situation and it is therefore interesting to examine the effects of replication under both these conditions.

4.2.4 Site Policies.

Different grid sites are likely to prioritise different kinds of job. A university with strong involvement in a particular experiment, for example, may prefer to accept jobs from that experiment, whereas a regional Tier 2 centre may be contracted to serve all experiments. In OptorSim, each site is given a list of job types which it will accept.

4.3 Optimisation Algorithms

There are two kinds of optimisation algorithm which may be investigated using OptorSim: the job scheduling algorithms used by the RB to decide which sites jobs should be sent to, and the data replication algorithms used by the RO at each site to decide when and how to replicate files. The focus of this paper is on the data replication algorithms, and so the job scheduling algorithms are not described here.

There are three broad options for replication strategies in OptorSim. Firstly, one can choose to perform no replication. Secondly, one can use a "traditional" algorithm which, when presented with a file request, always tries to replicate and, if necessary, deletes existing files to do so. Algorithms in this category are the LRU (Least Recently Used), which deletes those files which have been used least recently, and the LFU (Least Frequently Used), which deletes those which have been used least frequently in the recent past. Thirdly, one can use an economic model in which sites "buy" and "sell" files using an auction mechanism, and will only delete files if they are less valuable than the new file. Details of the auction mechanism and file value prediction algorithms can be found in [4]. There are currently two versions of the economic model: the binomial economic model, where file values are predicted by ranking the files in a binomial distribution according to their popularity in the recent past, and

[1] A Zipf-like distribution is defined as $P_i \propto i^{-\alpha}$, where P_i is the frequency of occurrence of the i^{th} ranked item and $\alpha \leqslant 1$ (a pure Zipf distribution would have $\alpha = 1$).

the Zipf economic model, where a Zipf-like distribution is used instead (a Zipf distribution is a power law in which a few events occur very frequently, while most events occur infrequently).

4.4 Evaluation Metrics

For evaluating grid performance, different users may have different criteria. An ordinary user will most likely be interested in the time a job takes to complete. Resource owners, on the other hand, will want to see their resources being used efficiently. The evaluation metric used in this paper are: *mean job time*, which is the average time a job takes to run, from the time of scheduling to completion; and *effective network usage (ENU)*, which is defined as

$$ENU = \frac{N_{remote\ file\ accesses} + N_{replications}}{N_{remote\ file\ accesses} + N_{local\ file\ accesses}}$$

The ENU is thus the ratio of file requests which use network resources to the total number of file requests, and so the lower the ENU the less loaded the network is and hence the more efficiently it is being used.

5 Experimental Setup

To evaluate the performance of these replication strategies in a realistic grid scenario, a simulation of LCG using the planned resources for physics analysis in 2008 was set up as follows. The topology, based on the the layout of the national research networks, is shown in Figure 2. The network capacity shown in Figure 2 was modified through the simulation by the addition of background network traffic, which varied according to the time of day in the simulation as shown in Figure 3. This profile was based on measurements of available bandwidth between CERN and Lyon, and assumes that patterns of network usage will not change significantly over the next few years.

5.1 Jobs and Files

Four datasets were defined, corresponding to the four major LHC experiment collaborations. Each of these datasets was placed at the Tier-0 (CERN) and each Tier-1 at the beginning of the simulation, so each file initially had 12 replicas around the grid. Six job types were defined, with the job and file parameters as shown in Table 1. The simulated jobs processed their given subset of files from the dataset. When a job ran on a site, it retrieved the files it required (according to the chosen access pattern) and processed them according to the computing resources available at that site. The probability

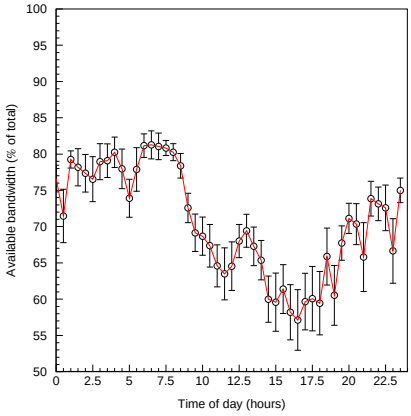


Figure 3: Profile of bandwidth variation used in LCG 2008 simulation.

of a particular job being run on the grid was modelled by the relative number of expected users for the different experiments.

Job	Dataset size (TB)	Total no. of files	Files/ job
`alice-pp`	50	25000	25
`alice-hi`	25	12500	125
`atlas`	200	100000	50
`cms`	75	37500	25
`lhcb-small`	75	37500	38
`lhcb-big`	75	37500	375

Table 1: Job configuration parameters used in the LCG 2008 configuration.

5.2 Resources

The compute and storage requirements for LCG in the first few years of LHC data-taking were drawn from the LCG Technical Design Report [2]. While the data is initially stored at Tier-0 and Tier-1 sites, user analysis jobs are expected to run at Tier-2 sites. Each Tier-2 was therefore given an an averaged CE of 645 kSI2000.The Tier-0 and Tier-1 sites were given SEs according to their planned capacities, each being sufficient to hold a complete copy of all the data files.

Each Tier-2 site was given a canonical value, averaging the total Tier-2 requirements over the number of Tier-2 sites. This gave an average SE size of 197 TB. Due to the limitations of available memory when running the simulation, however, the simulations were restricted to the order of 1000 jobs. To reach a state in the simulation were the SEs are full

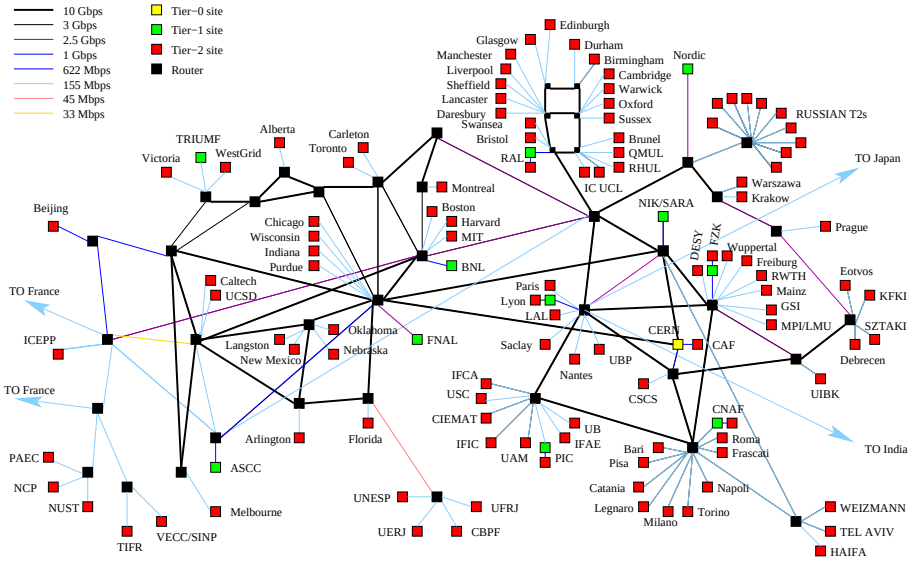


Figure 2: Simplified topology of LCG in 2008.

and file deletion is occurring would require about 200,000 jobs if the SEs were kept at 197 TB, and so the Tier-2 SE sizes were scaled down to 500 GB. These then hold 250 files, allowing file replacement to start when at most 10 jobs have been submitted to a site. This has the disadvantage that the file prediction algorithms will not perform to their best advantage. The effect of changing the size of the dataset compared to the SE sizes, however, is among the tests presented.

6 Results

For the results presented here, 1000 jobs were submitted to the grid at random intervals and, unless stated, a sequential access pattern was used. A job scheduling algorithm was used which balanced the queue length at sites with the data requirements of the jobs, which has previously been shown [7] to perform well. Tests were repeated at least 3 times and a mean value taken.

6.1 Varying Dataset Size

The ratio of average SE size to total dataset size can be a useful characterisation metric for a data grid, and is here designated D. The value of D indicates the likelihood of replication occurring. If $D > 1$, an average SE is able to hold all the files that jobs could require, so there will never be any deletion. For $D < 1$, the replication strategy becomes more important, as the SE is not capable of holding all the files and deletion must take place. For $D << 1$ due to a large dataset, however, replication will begin to lose its advantage, as each new job is likely to request files which are not in the access history. The value of D at which the switch between these two behaviours occurs will depend on the grid itself.

The default value for the experiments presented here, due to the scaling of Tier-2 SEs, is 1.2×10^{-3}, which is too small to effectively compare the replication algorithms. The first experiment presented therefore examines the dependency of the replication strategies on D. The overall dataset size was successively halved, varying D from 1.2×10^{-3} to 7.5×10^{-2}, bringing it closer to a more realistic level of $\mathcal{O}(10^{-1})$. The results of this test are shown in Figure 4. It did not prove possible to test higher values of D due to the memory limitations of the available hardware, but testing with $D \sim 1$ would be desirable in future work.

Firstly, as should be expected, the mean job time without replication is independent of D. For $D \lesssim 10^{-2}$, replication gives no advantage; for $D > 10^{-2}$, the mean job time drops rapidly for all the replication strategies. For the highest value of D tested, the LRU and LFU are slightly faster than the economic models. Examining the ENU, it is seen to fall as D increases, for all the replication algorithms. Without replication it naturally remains constant. This shows the increasing effectiveness of the replication strategies as their access histories contain a more representative sample of the whole dataset.

Although these results were gained with 1000 grid jobs, tests with simpler grids [11] show, within errors, a linear increase in job time with number of jobs up to $\mathcal{O}(10^5)$ jobs, which is more realistic. This means that with realistic values for D and higher

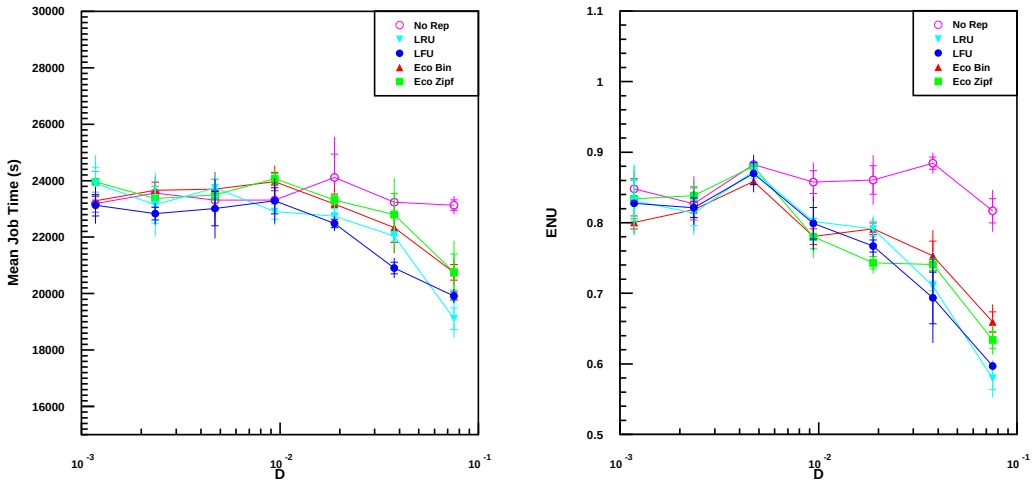


Figure 4: Mean job time (left) and ENU (right) for different replication algorithms, varying the value of D.

numbers of jobs, this relative improvement would hold. Replication is therefore an important way of reducing job times and network usage, and the relatively simple LRU and LFU strategies are the most effective for this topology.

In the sections which follow, the results are taken with the low value of D and it should therefore be remembered that in reality, replication would have a much stronger effect.

6.2 Effects of Site Policies

In the previous experiments, site policies - the types of jobs which would be accepted by each site - were set according to their planned usage. Here, the effect of site policies on the overall running of the grid are investigated. This was done by defining two extremes of policy. In the first, designated *All Job Types*, all sites accepted all job types. In the second, designated *One Job Type*, each site would accept only one job type, with an even distribution of sites for each job type. The default set of site policies is therefore in between these two extremes, and is designated in the results below as *Mixed*. The results are shown in Figure 5. These results show that the overall pattern of site policies on the grid have a powerful effect on performance. The mean job time with the *All Job Types* policy is about 60% lower than with the *One Job Type* policy. This is true across all the replication strategies, although the effect is strongest with no replication and with the LRU; it is again seen that the simple replication strategies are slightly faster than the economic models. *All Job Types* also gives a lower ENU (about 25% lower than the others). It seems clear that an egalitarian approach, in which resources are shared as much as possible, yields benefits to all grid users.

6.3 Effects of Zipf-like File Access

In all the experiments presented so far, jobs have accessed their files using a sequential access pattern. As Section 4.2.3 mentioned, however, Zipf-like file access may also be significant in a grid such as LCG. The performance of the replication strategies with a Zipf access pattern were therefore compared to that with sequential access, with the results shown in Figure 6. This shows quite a different pattern to the previous results. Although the four replication algorithms still have very similar performances, they are now about 75% faster than without replication. The ENU is correspondingly lower. This is due to the way in which a few files from each job's fileset are accessed many times during the jobs, while others are accessed infrequently. This allows the access histories to predict file values more accurately than with the sequential pattern, where they may see a file only once. As the number of jobs and the proportion of the whole dataset seen by an individual SE increases, however, the results with sequential access should tend towards a similar pattern as for the Zipf access. This is borne out by the results from varying D with sequential access. The presence of any Zipf-like element, even if combined with a sequential pattern, would make dynamic replication highly desirable.

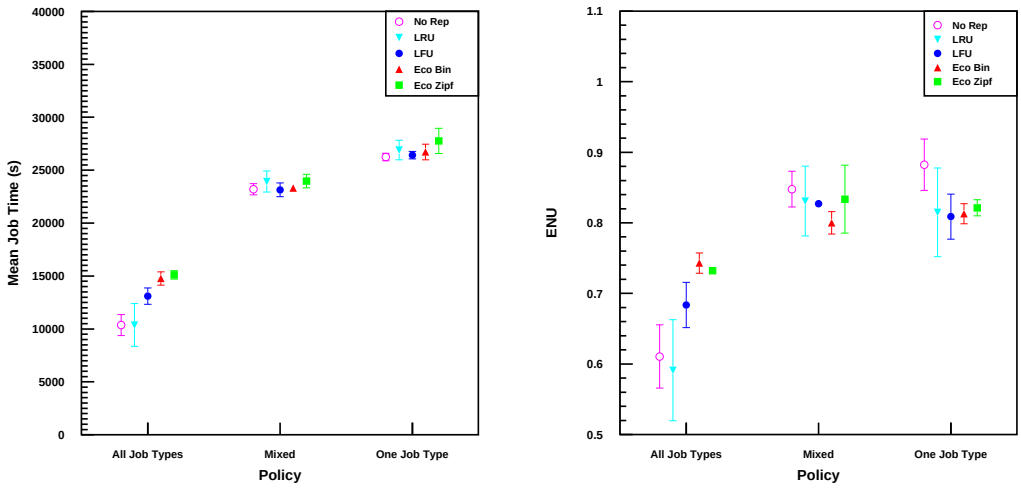


Figure 5: Mean job time (left) and ENU (right) for different replication algorithms, with different site policies.

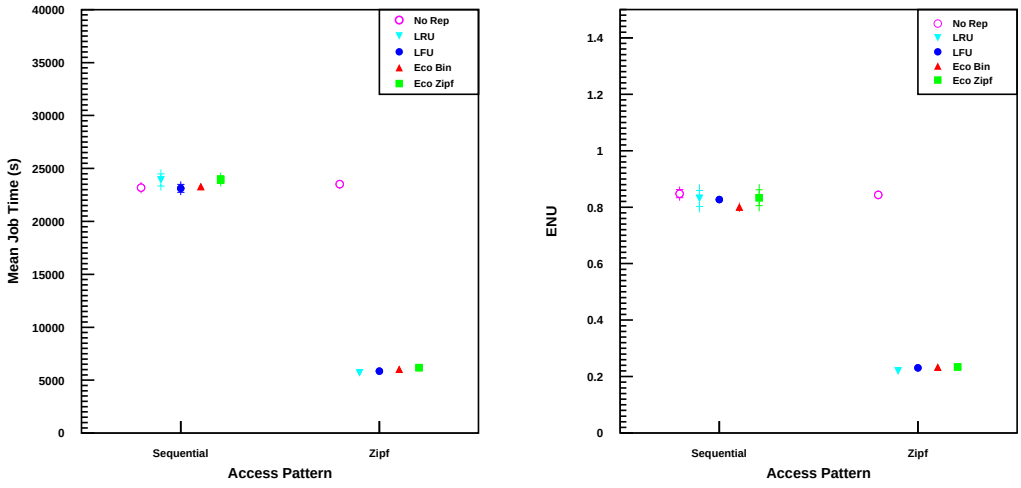


Figure 6: Mean job time (left) and ENU (right) for different replication algorithms, with Zipf-like access pattern.

7 Conclusions

OptorSim has been used to explore file replication strategies, under various conditions, in simulations of the LHC Computing Grid (LCG). LCG is a large and complex high-energy physics data grid, with well-defined resource plans and use cases, and so is a good test case for dynamic data replication. While the results presented here are for particle physics, OptorSim could also be used to simulate other data-intensive grids, such as those developed for biomedical or astronomical applications.

First, it was shown that dynamically replicating data between sites using a sequential file access pattern decreased the running time of grid jobs by about 20% and reduced usage of the network by about 25%, especially as sites' Replica Optimisers gained more knowledge of the overall dataset. While the performances of different replication strategies were similar, the simpler LRU and LFU strategies were found to perform up to 20% and 30% better, respectively, than the economic models. In previous work ([8], [7], [6]), the economic models were found to perform better; this may be due to the fact that in those scenarios, the grids were much simpler, leading to lower overheads for the economic model. Another reason may be that there were fewer initial replicas in these simulations, whereas in this case the higher number reduced the need for dynamic replication and thus reduced the gain from the economic models. In future, it would be good to extend the simulation to higher numbers of jobs and ascertain whether this behaviour holds.

Examining site policies, it was found that a policy which allowed all experiments to share site resources was most effective in reducing data access time and network usage. Finally, it was shown that if data access patterns are Zipf-like, dynamic replication has a much stronger effect than with sequential access, with performance gains of about 75%.

In general, dynamic replication is likely to be a valuable tool in improving grid performance under a range of conditions, not only for high energy physics grids but for any grid where large datasets are involved. While economic models of file replication may be useful for future grids, simple replication strategies such as LRU and LFU perform well for current-generation grids like LCG. The next step is to implement prototype tools for dynamic replication for testing in a real grid environment.

References

[1] OptorSim - A Replica Optimiser Simulation. `http://cern.ch/edg-wp2/optimization/optorsim.html`.

[2] LHC Computing Grid Technical Design Report. Technical Report CERN-LHCC-2005-024, CERN, June 2005.

[3] W. H. Bell, D. G. Cameron, L. Capozza, P. Millar, K. Stockinger, and F. Zini. Simulation of Dynamic Grid Replication Strategies in OptorSim. *Int. Journal of High Performance Computing Applications*, 17(4), 2003.

[4] W.H. Bell, D.G. Cameron, R. Carvajal-Schiaffino, A.P. Millar, K. Stockinger, and F. Zini. Evaluation of an Economy-Based File Replication Strategy for a Data Grid. In *Proceedings of 3nd IEEE Int. Symposium on Cluster Computing and the Grid (CCGrid 2003)*, Tokyo, Japan, May 2003. IEEE CS-Press.

[5] Rajkumar Buyya and Manzur Murshed. GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing. *Concurrency and Computation: Practice and Experience*, 14:1175–1220, 2002.

[6] D. Cameron, P. Millar, C. Nicholson, R. Carvajal-Schiaffino, F. Zini, and K. Stockinger. OptorSim: a Simulation Tool for Scheduling and Replica Optimisation in Data Grids. In *Computing in High Energy and Nuclear Physics (CHEP)*, Interlaken, Switzerland, September 2004.

[7] D. G. Cameron, R. Carvajal-Schiaffino, A. P. Millar, C. Nicholson, K. Stockinger, and F. Zini. UK Grid Simulation with OptorSim. In *UK e-Science All Hands Meeting*, Nottingham, UK, 2003.

[8] D. G. Cameron, R. Carvajal-Schiaffino, A. P. Millar, C. Nicholson, K. Stockinger, and F. Zini. Analysing Scheduling and Replica Optimisation Strategies for Data Grids with OptorSim. *Journal of Grid Computing*, 2(1):57–69, 2004.

[9] A. Iamnitchi and M. Ripeanu. Myth and reality: Usage behavior in a large data-intensive physics project. Technical Report TR2003-4, GriPhyN, 2003.

[10] Iosif Legrand. Multi-threaded, discrete event simulation of distributed computing system. In *Computing in High Energy and Nuclear Physics (CHEP)*, Padova, Italy, February 2000.

[11] C. M. M. Nicholson. *File Management for HEP Data Grids.* PhD thesis, University of Glasgow, 2006.

[12] K. Ranganathan and I. Foster. Decoupling Computation and Data Scheduling in Distributed Data Intensive Applications. In *Proc. of the 11th International Symposium for High Performance Distributed Computing (HPDC-11)*, Edinburgh, Scotland, 2002.

[13] A. Takefusa, O. Tatebe, S. Matsuoka, , and Y. Morita. Performance Analysis of Scheduling and Replication Algorithms on Grid Datafarm Architecture for High-Energy Physics Applications. In *Proc. of the 12th IEEE International Symposium on High Performance Distributed Computing (HPDC-12), IEEE Press*, June 2002.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Optimisation of Grid Enabled Storage at Small Sites

Greig A Cowan
University of Edinburgh, UK

Jamie K Ferguson, Graeme A Stewart
University of Glasgow, UK

Abstract

Grid enabled storage systems are a vital part of data processing in the grid environment. Even for sites primarily providing computing cycles, local storage caches will often be used to buffer input files and output results before transfer to other sites. In order for sites to process jobs efficiently it is necessary that site storage is not a bottleneck to Grid jobs.

dCache and DPM are two Grid middleware applications that provide disk pool management of storage resources. Their implementations of the storage resource manager (SRM) webservice allows them to provide a standard interface to this managed storage, enabling them to interact with other SRM enabled Grid middleware and storage devices. In this paper we present a comprehensive set of results showing the data transfer rates in and out of these two SRM systems when running on 2.4 series Linux kernels and with different underlying filesystems.

This benchmarking information is very important for the optimisation of Grid storage resources at smaller sites, that are required to provide an SRM interface to their available storage systems.

1 Introduction

1.1 Small sites and Grid computing

The EGEE project [1] brings together scientists and engineers from 30 countries to order to create a Grid infrastructure that is constantly available for scientific computing and analysis.

The aim of the Worldwide LHC Computing Grid (WLCG) is to use the EGEE developed software to construct a global computing resource that will enable particle physicists to store and analyse particle physics data that the Large Hadron Collider (LHC) and its experiments will generate when it starts taking data in 2007. The WLCG is based on a distributed Grid computing model and will make use of the computing and storage resources at physics laboratories and institutes around the world. Depending on the level of available resources, the institutes are organised into a hierarchy starting with the Tier-0 centre at CERN (the location of the LHC), multiple national laboratories (Tier-1 centres) and numerous smaller research institutes and Universities (Tier-2 sites) within each participating country. Each Tier is expected to provide a certain level of computing service to Grid users once the WLCG goes into full production.

The authors' host institutes form part of ScotGrid [2], a distributed WLCG Tier-2 centre, and it is from this point of view that we approach the subject of this paper. Although each Tier-2 is unique in its configuration and operation, similarities can be easily identified, particularly in the area of data storage:

1. Typically Tier-2 sites have limited hardware resources. For example, they may have one or two servers attached to a few terabytes of disk, configured as a RAID system. Additional storage may be NFS mounted from another disk server which is shared with other non-Grid users.
2. No tape storage.
3. Limited manpower to spend on administering and configuring a storage system.

The objective of this paper is to study the configuration of a Grid enabled storage element at a typical Tier-2 site. In particular we will investigate how changes in the disk server filesystems and file transfer parameters affect the data transfer rate when writing into the storage element. We concentrate on the case of writing to the storage element, as this is expected to be the most stressful operation on the Tier-2's storage resources, and indeed testing within the GridPP

collaboration bears this out [3]. Sites will be able to use the results of this paper to make informed decisions about the optimal setup of their storage resources without the need to perform extensive analysis on their own.

Although in this paper we concentrate on particular SRM [4] grid storage software, the results will be of interest in optimising other types of grid storage at smaller sites.

This paper is organised as follows. Section 2 describes the grid middleware components that were used during the tests. Section 3 then goes on to describe the hardware, which was chosen to represent a typical Tier-2 setup, that was used during the tests. Section 3.1 details the filesystem formats that were studied and the Linux kernel that was used to operate the disk pools. Our testing method is outlined in Section 4 and the results of these tests are reported and discussed in Section 5. In Section 6 we present suggestions of possible future work that could be carried out to extend our understanding of optimisation of Grid enabled storage elements at small sites and conclude in Section 7.

2 Grid middleware components

2.1 SRM

The use of standard interfaces to storage resources is essential in a Grid environment like the WLCG since it will enable interoperation of the heterogeneous collection of storage hardware that the collaborating institutes have available to them. Within the high energy physics community the storage resource manager (SRM) [4] interface has been chosen by the LHC experiments as one of the baseline services [5] that participating institutes should provide to allow access to their disk and tape storage across the Grid. It should be noted here that a storage element that provides an SRM interface will be referred to as 'an SRM'. The storage resource broker (SRB) [6] is an alternative technology developed by the San Diego Supercomputing Center [7] that uses a client-server approach to create a logical distributed file system for users, with a single global logical namespace or file hierarchy. This has not been chosen as one of the baseline services within the WLCG.

2.2 dCache

dCache [8] is a system jointly developed by DESY and FNAL that aims to provide a mechanism for storing and retrieving huge amounts of data among a large number of heterogeneous server nodes, which can be of varying architectures (x86, ia32, ia64). It provides a single namespace view of all of the files that it manages and allows access to these files using a variety of a protocols, including SRM. By connecting dCache to a tape backend, it becomes a hierarchical storage manager. However, this is not of particular relevance to Tier-2 sites who do not typically have tape storage. dCache is a highly configurable storage solution and can be easily deployed at Tier-2 sites where DPM is not sufficiently flexible.

2.3 Disk pool manager

Developed at CERN, and now part of the gLite middleware set, the disk pool manager (DPM) is similar to dCache in that is provides a single namespace view of all of the files stored on the multiple disk servers that it manages and provides a variety of methods for accessing this data, including SRM. DPM was always intended to be used primarily at Tier-2 sites, so has an emphasis on ease of deployment and maintenance. Consequently it lacks some of the sophistication of dCache, but is simpler to configure and run.

2.4 File transfer service

The gLite file transfer service (FTS) [9] is a grid middleware component that aims to provide reliable file transfer between storage elements that provide the SRM or GridFTP [10] interface. It uses the concept of channels [11] to define unidirectional data transfer paths between storage elements, which usually map to dedicated network pipes. There are a number of transfer parameters that can be modified on each of these channels in order to control the behaviour of the file transfers between the source and destination storage elements. We concern ourselves with two of these: the number of concurrent file transfers (N_f) and the number of parallel GridFTP streams (N_s). N_f is the number of files that FTS will simultaneously transfer in any bulk file transfer operation. N_s is number of simultaneous GridFTP channels that are opened up for each of these files.

2.5 Installation

Sections 2.2 and 2.3 described the two disk pool management applications that are suitable for use at small sites. Both dCache (v1.6.6-5) and DPM (v1.4.5) are available as part of the 2.7.0 release of the LCG software stack and have been sequentially investigated in the work presented here. In each case, the LCG YAIM [12] installation mechanism was used to create a default instance of the application on the available test machine (See Section 3). For dCache, PostGreSQL v8.1.3 was used, obtained from the PostGreSQL website [13]. Other than installing the SRM system in order that they be fully operational, no configuration options were altered.

3 Hardware configuration

In order for our findings to be applicable to existing WLCG Tier-2 sites, we chose test hardware representative of Tier-2 resources.

1. Single node with a dual core Xeon CPU. This operated all of the relevant services (i.e. SRM/nameserver and disk pool access) that were required for operation of the dCache or DPM.

2. 5TB RAID level-5 disk with a 64K stripe. Partitioned into three 1.7TB filesystems.

3. Source DPM for the transfers was a sufficiently high performance machine that it was able to output data across the network such that it would not act as a bottleneck during the tests.

4. 1Gb/s network connection between the source and destination SRMs, which were on the same network connected via a Netgear GS742T switch. During the tests, there was little or no other traffic on the network.

5. No firewalling (no iptables module loaded) between the source and destination SRMs.

3.1 Kernels and filesystems

Table 1 summarises the combinations of Linux kernels that we ran on our storage element and the disk pool filesystems that we tested it with. As can be seen four filesystems, ext2 [14], ext3 [15], jfs [16], xfs [17], were studied. Support for the first 3 filesystems is included by default in the Scientific Linux [18] 305 distribution. However, support for xfs is not enabled. In order to study the performance of xfs a CERN contributed rebuild of the standard Scientific Linux kernel was used. This differs from the first kernel only with the addition of xfs support.

Note that these kernel choices are in keeping with the 'Tier-2' philosophy of this paper – Tier-2 sites will not have the resources available to recompile kernels, but will instead choose a kernel which includes support for their desired filesystem.

In each case, the default options were used when mounting the filesystems.

	Filesystem			
Kernel	*ext2*	*ext3*	*jfs*	*xfs*
2.4.21	Y	Y	Y	N
2.4.21+cern xfs	N	N	N	Y

Table 1: Filesystems tested for each Linux kernel/distribution.

4 Method

The method adopted was to use FTS to transfer 30 1GB files from a source DPM to the test SRM, measuring the data transfer rate for each of the filesystem-kernel combinations described in Section 3.1 and for different values of the FTS parameters identified in Section 2.4. We chose to look at $N_f, N_s \in (1, 3, 5, 10)$. Using FTS enabled us to record the number of successful and failed transfers in each of the batches that were submitted. A 1GB file size was selected as being representative of the typical filesize that will be used by the LHC experiments involved in the WLCG.

Each measurement was repeated 4 times to obtain a mean. Any transfers which showed anomalous results (e.g., less than 50% of the bandwidth of the other 3) were investigated and, if necessary, repeated. This was to prevent failures in higher level components, e.g., FTS from adversely affecting the results presented here.

5 Results

5.1 Transfer Rates

Table 2 shows the transfer rate, averaged over N_s, for dCache for each of the filesystems. Similarly 3 shows the rate averaged over N_f.

N_f	Filesystem			
	ext2	*ext3*	*jfs*	*xfs*
1	157	146	137	156
3	207	176	236	236
5	209	162	246	245
10	207	165	244	247

Table 2: Average transfer rates per filesystem for dCache for each N_f.

N_s	Filesystem			
	ext2	*ext3*	*jfs*	*xfs*
1	217	177	234	233
3	191	159	214	219
5	189	155	208	217
10	183	158	207	215

Table 3: Average transfer rates per filesystem for dCache for each N_s.

Table 4 shows the transfer rate, averaged over N_s, for DPM for each of the filesystems. Similarly 5 shows the rate averaged over N_f.

The following conclusions can be drawn:

1. Transfer rates are greater when using modern high performance filesystems like jfs and xfs than the older ext2,3 filesystems.

2. Transfer rates for ext2 are higher than ext3, because it does not suffer a journalling overhead.

3. For all filesystems, having more than 1 simultaneous file transferred improves the average transfer rate substantially. There appears to be little dependence of the average transfer rate on the number of files in a multi-file transfer (for the range of N_f studied).

4a. With dCache, for all filesystems, single stream transfers achieve a higher average transfer rate than multistream transfers.

4b. With DPM, for ext2,3 single stream transfers achieve a higher average transfer rate than multistream transfers. For xfs and jfs multistreaming has little effect on the rate.

4c. In both cases, there appears to be little dependence of the average transfer rate on the number of streams in a multistream transfer (for the range of N_s studied).

N_f	Filesystem			
	ext2	*ext3*	*jfs*	*xfs*
1	214	192	141	204
3	297	252	357	341
5	300	261	368	354
10	282	253	379	356

Table 4: Average transfer rates per filesystem for DPM for each N_f.

N_s	Filesystem			
	ext2	*ext3*	*jfs*	*xfs*
1	293	277	289	310
3	264	209	313	323
5	264	237	303	307
10	272	234	339	317

Table 5: Average transfer rates per filesystem for DPM for each N_s.

5.2 Error Rates

Table 6 shows the average percentage error rates for the transfers obtained with both SRMs.

5.2.1 dCache

With dCache there were a small number of transfer errors for ext2,3 filesytems. These can be traced back to FTS cancelling the transfer of a single file. It is likely that the high machine load generated by the I/O traffic impaired the performance of the dCache SRM service. No errors were reported with xfs and jfs filesystems which can be correlated with the correspondingly lower load that was observed on the system compared to the ext2,3 filesystems.

5.2.2 DPM

With DPM all of the filesystems can lead to errors in transfers. Similarly to dCache these errors generally occur because of a failure to correctly call `srmSetDone()` in FTS. This can be traced back to the DPM SRM daemons being

SRM	Filesystem			
	ext2	*ext3*	*jfs*	*xfs*
dCache	0.21	0.05	0	0
DPM	0.05	0.10	1.04	0.21

Table 6: Percentage observed error rates for different filesystems and kernels with dCache and DPM.

badly affected by the machine load generated by the high I/O traffic. In general it is recommended to separate the SRM daemons from the actual disk servers, particularly at larger sites.

Note that the error rate for jfs was higher, by some margin, than for the other filesystems. However, this was mainly due to one single transfer which had a very high error rate and further testing should be done to see if this repeats.

5.3 Comment on FTS parameters

The poorer performance of multiple parallel GridFTP streams relative to a single stream transfer observed for dCache and for DPM with ext2,3 can be understood by considering the I/O behaviour of the destination disk. With multiple parallel streams, a single file is split into sections and sent down separate TCP channels to the destination storage element. When the storage element starts to write this data to disk, it will have to perform many random writes to the disk as different packets arrive from each of the different streams. In the single stream case, data from a single file arrives sequentially at the storage element, meaning that the data can be written sequentially on the disk, or at least with significantly fewer random writes. Since random writes involve physical movement of the disk and/or the write head, it will degrade the write performance relative to a sequential write access pattern.

In fact multiple TCP streams are generally beneficial when performing wide area network transfers, in order to maximise the network throughput. In our case as the source and destination SRMs were on the same LAN the effect of multistreams was generally detrimental.

It must be noted that a systematic study was not performed for the case of $N_f > 10$. However, initial tests show that if FTS is used to manage a bulk file transfer on a channel where N_f is initially set to a high value, then the destination storage element experiences a high load immediately after the first batch of files has been transferred, causing a corresponding drop in the observed transfer rate. This effect can be seen in Figure 1, where there is an initial high data transfer rate, but this reduces once the first batch of N_f files has been transferred. It is likely that the effect is due to the post-transfer negotiation steps of the SRM protocol occurring simultaneously for all of the transferred files. The resulting high load on the destination SRM node causes all subsequent FTS transfer requests to time out, resulting in the transfers failing. It must be noted that our use of a 1GB test file for

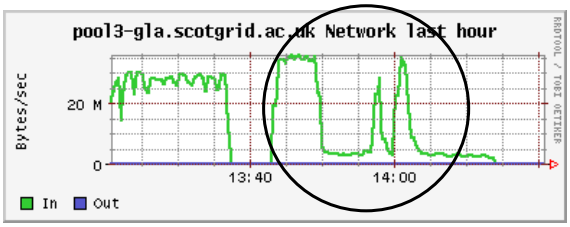


Figure 1: Network profile of destination dCache node with jfs pool filesystem during an FTS transfer of 30 1GB files. Throughout the transfer, $N_f = 15$. Final rate was 142Mb/s with 15 failed transfers.

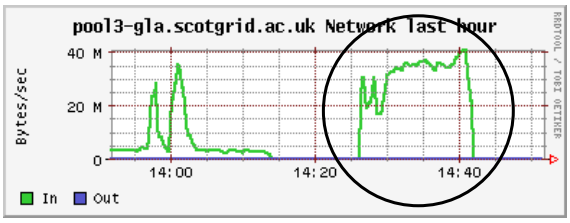


Figure 2: Network profile of destination dCache node with jfs pool filesystem during an FTS transfer of 30 1GB files. $N_f = 1$ at the start of the transfer, increasing up to $N_f = 15$ as the transfer progressed. Final transfer rate was 249Mb/s with no failed transfers.

all transfers will exacerbate this effect. Figure 2 shows how this effect disappears when the start times of the file transfers are staggered by slowly increasing N_f from $N_f = 1$ to 15, indicating that higher file transfer rates as well as fewer file failures could be achieved if FTS staggered the start times of the file transfers. Improvements could be made if multiple nodes were used to spread the load of the destination storage element. If available, separate nodes could be used to run the disk server side of the SRM and the namespace services.

6 Future Work

The work described in this paper is a first step into the area of optimisation of grid enabled storage at Tier-2 sites. In order to fully understand and optimise the interaction between the base operating system, disk filesystem and storage applications, it would be interesting to extend this research in a number of directions:

1. Using non-default mount options for each of the filesystems.

2. Repeat the tests with SL 4 as the base oper-

ating system (which would also allow testing of a 2.6 Linux kernel).

3. Investigate the effect of using a different stripe size for the RAID configuration of the disk servers.

4. Vanilla installations of SL generally come with unsuitable default values for the TCP read and write buffer sizes. In light of this, it will be interesting to study the how changes in the Linux kernel-networking tuning parameters change the FTS data transfer rates. Initial work in this area has shown that improvements can be made.

5. It would be interesting to investigate the effect of different TCP implementations within the Linux kernel. For example, TCP-BIC [19], westwood [20], vegas [21] and web100 [22].

6. Characterise the performance enhancement that can be gained by the addition of extra hardware, particularly the inclusion of extra disk servers.

When the WLCG goes into full production, a more realistic use case for the operation of an SRM will be one in which it is simultaneously writing (as is the case here) and reading files across the WAN. Such a simulation should also include a background of local file access to the storage element, as is expected to occur during periods of end user analysis on Tier-2 computing clusters. Preliminary work has already started in this area where we have been observing the performance of existing SRMs within ScotGrid during simultaneous read and writing of data.

7 Conclusion

This paper has presented the results of a comparison of file transfer rates that were achieved when FTS was used to copy files between Grid enabled storage elements that were operating with different destination disk pool filesystems and a 2.4 Linux kernel. Both dCache and DPM were considered as destination disk pool managers providing an SRM interface to the available disk. In addition, the dependence of the file transfer rate on the number of concurrent files (N_f) and the number of parallel GridFTP streams (N_s) was investigated.

In terms of optimising the file transfer rate that could be achieved with FTS when writing into a characteristic Tier-2 SRM setup, the results can be summarised as follows:

1. Pool filesystem: jfs or xfs.

2. FTS parameters: Low value for N_s, high value for N_f (staggering the file transfer start times). In particular, for dCache $N_f = 1$ was identified as giving optimal transfer rate.

It is not possible to make a single recommendation on the SRM application that sites should use based on the results presented here. This decision must be made depending upon the available hardware and manpower resources and consideration of the relative features of each SRM solution.

Extensions to this work were suggested, ranging from studies of the interface between the kernel and network layers all the way up to making changes to the hardware configuration. Only when we understand how the hardware and software components interact to make up an entire Grid enabled storage element will we be able to give a fully informed set of recommendations to Tier-2 sites regarding the optimal SRM configuration. This information is required by them in order that they can provide the level of service expected by the WLCG and other Grid projects.

8 Acknowledgements

The research presented here was performed using ScotGrid [2] hardware and was made possible with funding from the Particle Physics and Astronomy Research Council through the GridPP project [23]. Thanks go to Paul Millar and David Martin for their technical assistance during the course of this work.

References

[1] Enabling Grids for E-scienceE
http://www.eu-egee.org/

[2] ScotGrid, the Scottish Grid Service
http://www.scotgrid.ac.uk

[3] GridPP Service Challenge Transfer tests
http://wiki.gridpp.ac.uk/wiki
/Service_Challenge_Transfer_Tests

[4] The SRM collaboration
http://sdm.lbl.gov/srm-wg/

[5] LCG Baseline Services Group Report *http://cern.ch/LCG/peb/bs/BSReport-v1.0.pdf*

[6] The SRB collaboration *http://www.sdsc.edu/srb/index.php/Main_Page*

[7] The San Diego Supercomputing Center *http://www.sdsc.edu*

[8] dCache collaboration *http://www.dcache.org*

[9] gLite FTS *https://uimon.cern.ch/twiki/bin/view/LCG/FtsRelease14*

[10] Data transport protocol developed by the Globus Alliance *http://www.globus.org/grid_software/data/gridftp.php*

[11] P. Kunszt, P. Badino, G. McCance, The gLite File Transfer Service: Middleware Lessons Learned from the Service Challenges, CHEP 2006 Mumbai. *http://indico.cern.ch/contributionDisplay.py?contribId=20&sessionId=7&confId=048*

[12] LCG generic installation manual *http://grid-deployment.web.cern.ch/grid-deployment/documentation/LCG2-Manual-Install/*

[13] PostGreSQL Database website *http://www.postgresql.org/*

[14] R.Card, T. Ts'o, S. Tweedie (1994). "Design and implementation of the second extended filesystem." Proceedings of the First Dutch International Symposium on Linux. ISBN 90-367-0385-9.

[15] Linux ext3 FAQ *http://batleth.sapienti-sat.org/projects/FAQs/ext3-faq.html*

[16] Jfs filesystem homepage *http://jfs.sourceforge.net/*

[17] Xfs filesystem homepage *http://oss.sgi.com/projects/xfs/index.html*

[18] Scientific Linux homepage *http://scientificlinux.org*

[19] A TCP variant for high speed long distance networks *http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/index.htm*

[20] TCP Westwood details *http://www.cs.ucla.edu/NRL/hpi/tcpw/*

[21] TCP Vegas details *http://www.cs.arizona.edu/protocols/*

[22] TCP Web100 details *http://www.hep.ucl.ac.uk/~ytl/tcpip/web100/*

[23] GridPP - the UK particle physics Grid *http://gridpp.ac.uk*

Profiling OGSA-DAI Performance for Common Use Patterns

Bartosz Dobrzelecki[1], Mario Antonioletti[1], Jennifer M. Schopf[2,3], Alastair C. Hume[1], Malcolm Atkinson[2], Neil P. Chue Hong[1], Mike Jackson[1], Kostas Karasavvas[2], Amy Krause[1], Mark Parsons[1], Tom Sugden[1], and Elias Theocharopoulos[2]

1. EPCC, University of Edinburgh, JCMB, The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, UK.
2. National e-Science Centre, University of Edinburgh, Edinburgh EH8 9AA, UK.
3. Distributed Systems Laboratory, Argonne National Laboratory, Argonne, IL, 60439 USA.

Abstract

OGSA-DAI provides an extensible Web service-based framework that allows data resources to be incorporated into Grid fabrics. The current OGSA-DAI release (OGSA-DAI WSI/WSRF v2.2) has implemented a set of optimizations identified through the examination of common OGSA-DAI use patterns. In this paper we describe these patterns and detail the optimizations that have been made for the current release based on the profiles obtained. These optimizations include improvements to the performance of various data format conversion routines, the introduction of more compact data delivery formats, and the adoption of SOAP with attachments for data delivery. We quantify the performance improvements in comparison to the previous OGSA-DAI release.

1. Introduction

The *Open Grid Services Architecture – Data Access and Integration* (OGSA-DAI) project [OD] aims to provide the e-Science community with a middleware solution to assist with the access and integration of data for applications working within Grids. Early Grid applications focused principally on the storage, replication, and movement of file-based data, but many of today's applications need full integration of database technologies with Grid middleware. Not only do many Grid applications already use databases for managing metadata, but increasingly many are associated with large databases of domain-specific information, for example, biological or astronomical data.

OGSA-DAI offers a collection of services for adding database access and integration capabilities to the core capabilities of service-oriented Grids, thus allowing structured data resources to be integrated with Grid applications. A systematic performance analysis of OGSA-DAI v2.1 [AAB+05] led to the emphasis on performance for the v2.2 release.

Our current work has focused on identifying common use patterns and their associated bottlenecks and then improving the performance of these particular use cases. The optimizations implemented include speeding data format conversion routines, introducing more compact data delivery formats, and using SOAP with attachments for data delivery. Section 2 references more detailed descriptions of OGSA-DAI and outlines related middleware and performance studies. Section 3 describes the use patterns adopted to identify the performance bottlenecks and describes the performance enhancements made. Section 4 quantifies the improvements resulting from these changes, comparing the performance of the present WSRF OGSA-DAI release v2.2, with the previous v2.1 release. Section 5 presents some conclusions derived from this work.

2. Related Work

Various publications provide information about the design of OGSA-DAI [AAB+05a], ways in which it can be used [KAA+05], the related WS-DAI family of specifications [AKP+06], and details about the OGSA-DQP software [AMP+03] which adds distributed query-processing capabilities through OGSA-DAI. Up-to-date documentation, tutorials, and

downloads related to OGSA-DAI are available from [OD].

2.1. Related Applications

Three current projects have closely related functionality to that of OGSA-DAI: the Storage Resource Broker, WebSphere Information Integrator, and Mobius.

The Storage Resource Broker (SRB) [SRB], developed by the San Diego Supercomputer Center, provides access to *collections*, or sets of data objects, using attributes or logical names rather than their physical names or locations. SRB is primarily file oriented but can also work with data objects, including archival systems, binary large objects in a database management system, database objects that may be queried by using SQL, and tape library systems. By contrast, OGSA-DAI takes a database oriented approach which also includes access to files. These two approaches are generally suited to differing problems, but SRB and OGSA-DAI can complement each other.

WebSphere Information Integrator (WSII), a commercial product from IBM, is commonly used to search data spanning organisational domains, data federation and replication, data transformation, and data event publishing [WSII]. Data federation allows multiple data sources to be queried and accessed through a single access point. IBM recently developed a Grid wrapper for WSII using OGSA-DAI, taking advantage of its abstraction capabilities, to wrap additional data resources that WSII can then access [LMD+05]. A more detailed comparison between OGSA-DAI and WSII can be found in [SH05].

Mobius [Mob], developed at Ohio State University, provides a set of tools and services to facilitate the management and sharing of data and metadata in a Grid. In order to expose a resource in Mobius, the resource must be described by using an XML Schema, which is then shared via their Global Model Exchange. The resource can then be accessed by querying the Schema using, for example, XPath. OGSA-DAI, in contrast, does not require an XML Schema to be created for a resource; rather, it directly exposes that information (data and metadata/schema) and is queried by using the resource's intrinsic querying mechanisms.

Several other projects, including ELDAS [ELD] and Spitfire [SF], also address the use of databases in a Grid environment but are not as commonly used.

2.2. Related Performance Studies

The Extreme Grid Web Services group at Indiana University have examined the performance of SOAP for high-performance and Grid computing [CGB02, GSC+00]. They have developed an *XML Pull Parser* implementation that is significantly faster than Xerces [Slo04]. This work concentrates on the XML processing and is not specific to database use patterns but one that potentially could be exploited by OGSA-DAI through its use of Web services.

A large body of work exists on benchmarking relational database systems. In particular, the Wisconsin Benchmark [Gra93] consists of queries that test the performance of small numbers of join operations. The XML Benchmark Project [SWK+01] has presented an approach to benchmarking XML databases [SWK+01a]. However, none of this work has looked at benchmarking Web service interfaces to databases – the area that OGSA-DAI occupies.

Some attempts have been made to use standard benchmarks to investigate the overheads of providing database access through Web service-style interfaces, for example by using TPC-H [HIM02]. This particular work focuses on network and encryption overhead but using a non SOAP-based interface. Nevertheless, this work is of interest for comparison for SOAP-based access.

Previous work has also been done to try to understand the performance of earlier versions of OGSA-DAI [JAC+03, AAB+05, AGM+05]. This work mainly looked at particular technical issues, whereas the work presented in this paper seeks to use common use patterns as a starting point for on-going optimisation.

3. Performance Bottlenecks in Common Use Patterns

OGSA-DAI employs Web services to expose intrinsic data resource capabilities and the data contained to its clients. In most instances, the data resources used are relational databases that require read-only access (query) and no write access (update/insert). For this reason, our study has focused on relational databases and used the execution of an SQL query as the base test case.

3.1. Use Case 1: Executing an SQL Query on a Remote Server

A typical OGSA-DAI client-service interaction involves a client running an SQL query through a remote OGSA-DAI service that then returns the query response, typically some data, in an XML document. This interaction involves the following six steps:

(1) The client sends a request containing the SQL query in a SOAP message to an OGSA-DAI service.
(2) The server extracts the request from the SOAP message, and the SQL query is executed on the relational database.
(3) The query results are returned from the relational database to the OGSA-DAI server as a set of Java ResultSet objects.
(4) The server converts the Java ResultSet objects into a format suitable for transmission back to the client, such as WebRowSet.
(5) This data is sent back to the client in a SOAP message.
(6) The client receives the SOAP message, unpacks the data, and converts it back to a ResultSet object (assuming this is a Java client).

3.1.1. Improvement 1: Faster Conversion

Profiling this use case showed that the conversion process, where a ResultSet object is converted to the WebRowSet format on the server (step 4), as well as the inverse process on the client (step 6), was the primary performance bottleneck, and an obvious area for improvement.

Previously, these converter routines were also applied to produce binary data, going from ResultSets to some suitable binary format. However, the cost of iteratively having to convert ResultSets to binary data proved to be too high. So, the converters were restricted to only deal with text based formats. This benefited the performance. In addition, a routine that used a regular expression Java API to escape XML special characters in data fields proved to be too expensive and was thus replaced by a more efficient parser that worked on arrays. These combined modifications improved the performance of the converters.

3.1.2. Improvement 2: Change in Data Format

In analyzing the overhead for our first use case, we also noticed that using WebRowSet as an intermediate delivery format added a significant amount of XML mark-up that increased the amount of data that needed to be transferred between the client and server, often up to twice the original size. In addition, the parsing of the messages out of XML could be slow.

This scenario also identified the fact that WebRowSet was only being used as an intermediate delivery format and was thus possibly incurring an unnecessary overhead. Hence, a second improvement we investigated was the use of an alternative intermediate delivery format, namely, *Comma Separated Values* (CSV). This format uses space more efficiently and is easier to parse, but it has two significant drawbacks. First, it provides only limited support for metadata – namely, an optional line with column names – so the embedding of metadata has weaker support than is the case with the WebRowSet format. Second, as there is no standard for representing relational data in CSV format, third-party tools may have difficulty interpreting OGSA-DAI-generated CSV files, despite

the fact that common conventions are used and they are internally consistent.

The reduction in data size in going from a WebRowSet to a CSV format can be estimated by calculating the space required to represent the same result in each format. This can be done by calculating the number of extra characters needed to describe a row of data. Assuming for CVS data that all fields are wrapped in double quotes and that there are no escaped characters, then the extra number of characters needed to represent a row in CSV document is: (number_of_columns * 3) – two quotes and a comma – whereas for WebRowSet the use of specific WebRowSet defined XML tags make this number: (number_of_columns * 27)+25. So, WebRowSet always requires at least nine time as many non-data characters as CSV.

3.2. Use Case 2: Transferring Binary Data

A slight variation on the previous use case involves using OGSA-DAI to provide access to files stored on a server's file system. Commonly, these could be large binary data files (e.g., medical images), stored in a file system, with the associated metadata for these files stored separately in a relational database. The client queries the databases to locate any files of interest, which are then retrieved by using the OGSA-DAI delivery mechanisms. Files are retrieved separately from the SOAP interactions for data transport efficiency. In some cases, however, it can be more convenient for a client to receive the data back in a SOAP response message rather than using an alternative delivery mechanism.

The implementation of this scenario includes the same six steps as in the first use case except that step 4 now requires the conversion of a binary file to a text-based format, usually Base64 encoding, in order for it to be sent back in a SOAP message, and step 6 includes decoding of the file back into its original binary format.

3.2.1. Improvement 3: SOAP with Attachments

The major bottleneck arising from this scenario is the Base64 encoding of the binary data for inclusion in a SOAP message. This encoding requires additional computation at both the client and the server side. Moreover, the converted data is approximately 135% of the size of the original file, clearly impacting the efficiency of the data transfer.

We have addressed both of these concerns by using SOAP messages with attachments [BTN00]. This approach significantly reduces the time required to process SOAP messages and allows the transfer of binary data to take place without necessitating Base64 encoding. The one difficulty with this approach is that, as SOAP messages with attachments are not a standard feature of all SOAP specifications, interoperability issues can arise.

4. Experimental Results

To quantify the effect of the performance improvements outlined in the preceding section, we compared the performance of OGSA-DAI WSRF v2.2 with OGSA-DAI WSRF v2.1.

4.1. Experimental Setup

We ran our experiments using an Apache Tomcat 5.0.28 / Globus Toolkit WS-Core 4.0.1 stack. Our experimental setup consisted of a client machine and a server machine on the same LAN. We ran the server code on a Sun Fire V240 Server, which has a dual 1.5 GHz UltraSPARC IIIi processor and 8 GB of memory, running Solaris 10 with J2SE 1.4.2_05. The client machine was a dual 2.40GHz Intel Xeon system running Red Hat 9 Linux with the 2.4.21 kernel and J2SE 1.4.2_08. Both the client JVM and the JVM running the Tomcat container were started in server mode by using the `-server -Xms256m -Xmx256m` set of flags.

The network packets in these experiments traversed two routers, and iperf 1.7.0 found the network bandwidth to be approximately 94 Mbits/s. The average round-trip latency was less than 1 ms.

The database used in the experiments was MySQL 5.0.15 with MySQL Connector/J driver version 3.1.10. We used *littleblackbook*, the sample database table distributed with OGSA-DAI, for all experiments. The average row length for this table is 66 bytes. The rows have the following schema: int(11), varchar(64), varchar(128), varchar(20).

Before we took any measurements, both the client and server JVMs were warmed up. Then each test was executed 10 times. The results reported are the average of these runs, with error bars indicating +/- standard deviation.

4.2. Faster Conversions and Change in Data Format

Our first set of experiments was based on the two improvements suggested by our first use case, namely, optimizing the code to do the data format conversions faster and evaluating the use of CSV instead of WebRowSet for an intermediate format.

For a set of queries, returning results consisting from 32 to 16,384 rows, we measured the time to perform the 6 steps involved in a client-service interaction, including the translation of the results into (and out of) WebRowSet or CSV formats as appropriate.

Figure 1 shows the overall timing results. Queries for 512 rows or greater show a significant improvement, up to 35% by simply optimizing the WebRowSet conversion (Improvement 1). The use of CSV instead of WebRowSet (Improvement 2) also shows a significant improvement for larger queries, up to 65% over the original v2.1 and about 50% over simply optimizing the conversion.

Figure 2 shows in more detail the performance improvements on the server side using Apache Axis logging to obtain the times spent in the different phases of the SOAP request processing. We divide the server performance into three phases:

1. Axis Parsing: the time spent in Apache Axis parsing a SOAP request.

2. OGSA-DAI Server: the time OGSA-DAI spent performing the requested activities and building the response document.

3. Message Transfer: the time the server spent sending a message back to the client.

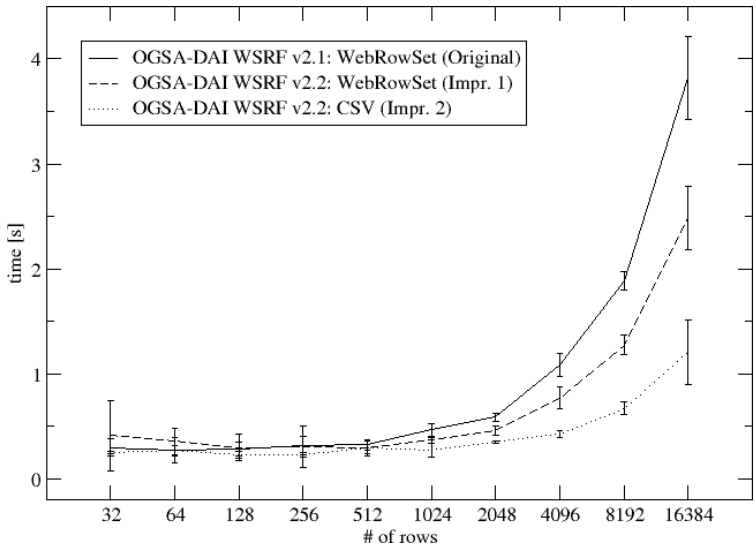


Figure 1: Measurements comparing the effects of better conversion code (Improvement 1) and using CSV formatting instead of WebRowSet format (Improvement 2) against the original v2.1 code. Results include both client and server times.

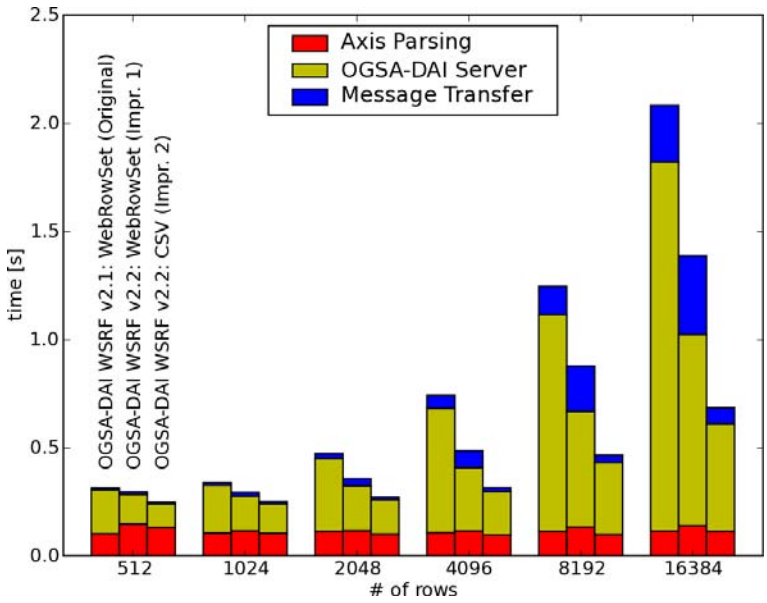


Figure 2: Time spent in the server only, split into three phases: Apache Axis parsing, OGSA-DAI server work, and the message transfer to the client.

The time spent in the Axis parsing phase is roughly constant as we always perform the same operation. In all cases the time spent in the OGSA-DAI server phase dominates and generally increases systematically with the size of the query results obtained. The largest portion of this phase is spent translating the ResultSet objects from a Java ResultSet object into WebRowSet or CSV. The optimised conversion to WebRowSet (Improvement 1) works up to 50% faster than the original version for large result sets. By using CSV instead of WebRowSet (Improvement 2), we also see a large reduction in delivery time and reduced network traffic, the Message Transfer phase.

4.3. Using SOAP Attachments

The second set of experiments we ran was to test the use of SOAP with attachments, as outlined in our second use case and Improvement 3 in Section 3.2. We compare using Byte64 encoding and returning the data in the body of a SOAP message to using SOAP messages with attachments to transfer a binary file.

Figure 3 shows the time taken to transfer binary data of increasing size using both delivery methods. We only have data for the Original v2.1 code up to 8MB file sizes because the process of Base64 encoding and building SOAP response for file sizes of 16MB upwards consumed all the heap memory available to the JVM and consequently caused the JVM to terminate. The performance gain shows nonlinear growth with increasing file size. Transferring an 8 MB file as a SOAP attachment takes only 25% of the time needed to transfer the same file inside the body of a SOAP message. This improvement is due to the fact that SOAP attachments do not need any special encoding, and less time is spent processing XML because the data is outside the body of the SOAP message.

Figure 4 gives additional detail about the server-side performance using the three previously defined phases. The Axis parsing phase is roughly constant, as before. During the OGSA-DAI server phase, the original SOAP delivery method is CPU bound because of the Base64 conversion required, while the performance of the new SOAP with attachments case is generally much better, limited mainly by the performance of the I/O operations rather than those of the CPU.

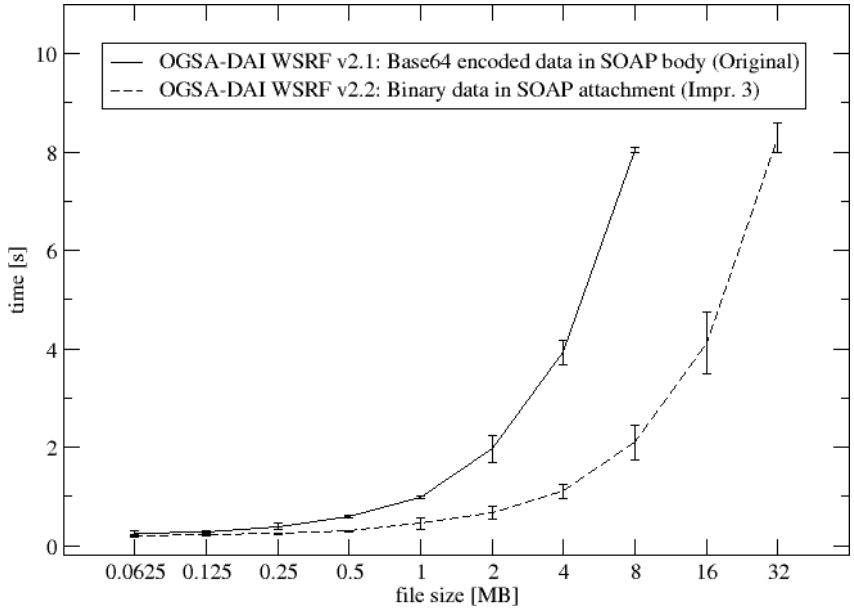


Figure 3: Time taken to transfer a binary file using Base64 encoded data inside the body of a SOAP message and as a SOAP attachment (Improvement 3).

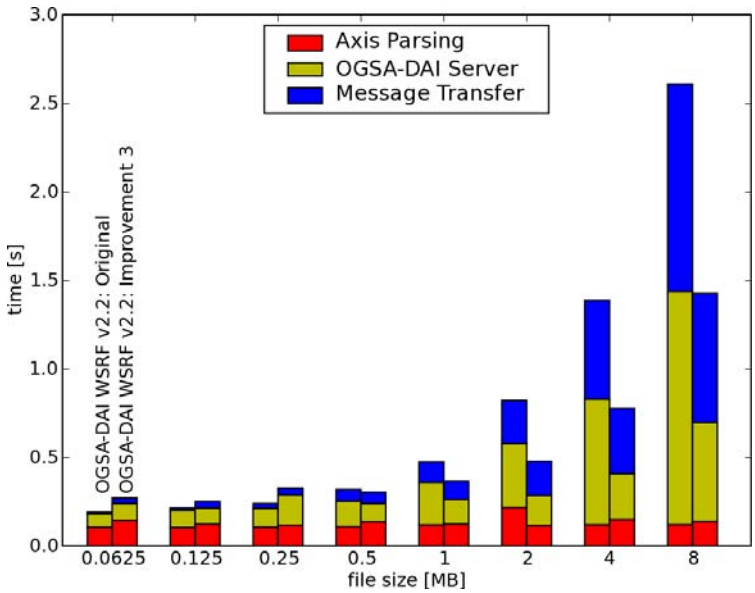


Figure 4: Time spent on server side split into phases. For each group the bar on the left measures the time spent sending binary data inside a SOAP message (Original) while the right bar corresponds to the approach where binary data is sent as a SOAP attachment (Improvement 3).

A similar performance gain is seen in the message transfer phase, where the absence of Base64 encoding reduces the quantity of data that needs to be transferred by up to 35% for the SOAP with attachments case.

4.4. SQL Results Delivery Using SOAP Attachments

The final experiment combined the previous two, by repeating the SQL query results retrieval from the first experiment but also using SOAP with attachments for data delivery. Figure 5 shows that there is little difference in the performance for

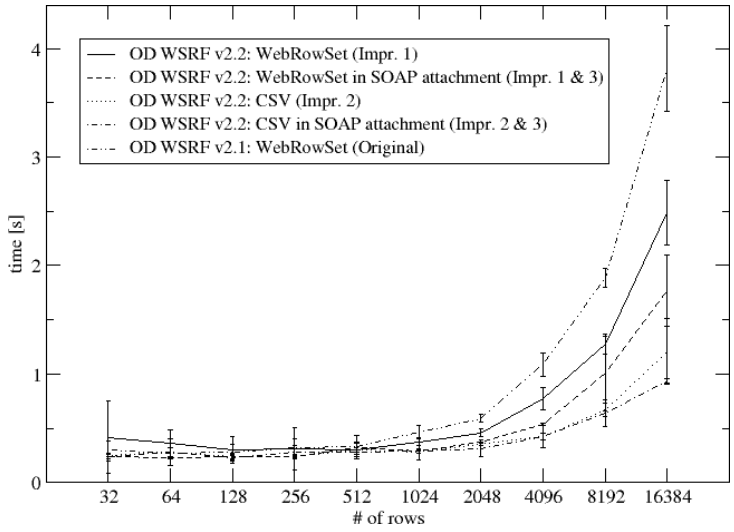


Figure 5: Execution time for scenarios fetching SQL results converted to XML and CSV data using two delivery mechanisms: delivery inside the body of a SOAP message and delivery as a SOAP attachment.

small data sets (less than 512 rows). For larger data sets, however, SOAP with attachments can achieve transfer times that are up to 30% faster than when using WebRowSet for delivery. Only the largest data set sees a performance gain when CSV formatting is used with SOAP attachments.

5. Conclusions

This paper summarises part of an ongoing effort to improve the performance of OGSA-DAI. We have analysed two typical use patterns, which were then profiled and the results used as a basis for implementing a focused set of performance improvements. The benefit of these has been demonstrated by comparing the performance of the current release of OGSA-DAI, which includes the performance improvements, with the previous release, which does not. We have seen performance improvements of over 50% in some instances. Source code and the results data are available from [Dob06].

Acknowledgements

This work is supported by the UK e-Science Grid Core Programme, through the Open Middleware Infrastructure Institute, and by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38.

We also gratefully acknowledge the input of our past and present partners and contributors to the OGSA-DAI project including: EPCC, IBM UK, IBM Corp., NeSC, University of Manchester, University of Newcastle and Oracle UK.

References

[AAB+05] M Antonioletti, M. Atkinson, R. Baxter, A Borley, N. P. Chue Hong, P. Dantressangle, A. C. Hume, M. Jackson, A. Krause, S. Laws, M. Parsons, N. W. Paton, J. M. Schopf, T. Sugden, P. Watson, and D. Vyvyan, OGSA-DAI Status and Benchmarks, Proceedings of the UK e-Science All Hands Meeting, 2005.

[AAB+05a] M. Antonioletti, M.P. Atkinson, R. Baxter, A. Borley, N.P. Chue Hong, B. Collins, N. Hardman, A. Hume, A. Knox, M. Jackson, A. Krause, S. Laws, J. Magowan, N.W. Paton, D. Pearson, T. Sugden, P. Watson, and M. Westhead. The Design and Implementation of Grid Database Services in OGSA-DAI. Concurrency and Computation:

Practice and Experience, Volume 17, Issue 2-4, Pages 357-376, February 2005.

[AGM+05] M.N. Alpdemir, A. Gounaris, A. Mukherjee, D. Fitzgerald, N. W. Paton, P. Watson, R. Sakellariou, A. A.A. Fernandes, and J. Smith, Experience on Performance Evaluation with OGSA-DQP, Proceedings of the UK e-Science All Hands Meeting, 2005.

[AKP+06] M. Antonioletti, A. Krause, N. W. Paton, A. Eisenberg, S. Laws, S. Malaika, J. Melton, and D. Pearson. The WS-DAI Family of Specifications for Web Service Data Access and Integration. ACM SIGMOD Record, Vol 35, No 1, pp48-55, 2006.

[AMP+03] M. N. Alpdemir, A. Mukherjee, N. W. Paton, P. Watson, A. A. A. Fernandes, A. Gounaris, and J. Smith. Service-Based Distributed Querying on the Grid. Service-Oriented Computing - ICSOC 2003 Editors: M. E. Orlowska, S. Weerawarana, M. P. Papazoglou, J. Yang. Lecture Notes in Computer Science, Volume 2910, pp. 467-482 Springer Berlin/Heidelberg 2003.

[BTN00] J. J Barton, S. Thatte, and H. F. Nielsen. *SOAP Messages with Attachments*. W3C Note 11 December 2000.

[CGB02] K. Chiu, M. Govindaraju, and R. Bramley, Investigating the Limits of SOAP Performance for Scientific Computing, Proceedings of HPDC 2002.

[Dob06] B. Dobrzelecki, Code and raw data from experiments, www.ogsadai.org.uk/documentation/scenarios/performance, 2006.

[ELD] ELDAS (Enterprise Level Data Access Services), EDIKT, www.edikt.org/eldas.

[Gra93] J. Gray, Database and Transaction Processing Performance Handbook. www.benchmarkresources.com/ handbook, 1993.

[GSC+00] M. Govindaraju, A. Slominski, V. Choppella, R. Bramley, and D. Gannon, On the Performance of Remote Method Invocation for Large-Scale Scientific Applications, Proceedings of SC'00, 2000.

[HIM02] H. Hacigumus, B. Iyer, and S. Mehrotra, Providing database as a service, Proceedings of 18th International Conference on Data Engineering 2002.

[JAC+03] M. Jackson, M. Antonioletti, N. Chue Hong, A. Hume, A. Krause, T. Sugden, and M. Westhead, Performance Analysis of the OGSA-DAI Software, Proceedings of the UK e-Science All Hands Meeting, 2003.

[KAA+05] K. Karasavvas, M. Antonioletti, M.P. Atkinson, N.P. Chue Hong, T. Sugden, A.C. Hume, M. Jackson, A. Krause, and C. Palansuriya. Introduction to OGSA-DAI Services. Lecture Notes in Computer Science, Volume 3458, Pages 1-12, May 2005.

[LMD+05] A. Lee, J. Magowan, P. Dantressangle, and F. Bannwart. Bridging the Integration Gap, Part 1: Federating Grid Data. IBM Developer Works, August 2005.

[Mob] Mobius, projectmobius.osu.edu.

[OD] Open Grid Services Architecture – Data Access and Integration (OGSA-DAI), www.ogsadai.org.uk.

[SH05] R. O. Sinnott and D. Houghton, Comparison of Data Access and Integration Technologies in the Life Science Domain, Proceedings of the UK e-Science All Hands Meeting 2005, September 2005.

[SF] Spitfire, edg-wp2.web.cern.ch/edg-wp2/spitfire.

[Slo04] A. Slominski. www.extreme.indiana.edu/~aslom/xpp_sax2bench/results.html, 2004.

[SRB] Storage Resource Broker (SRB), www.sdsc.edu/srb.

[SWK+01] A. R. Schmidt, Florian Waas, M. L. Kersten, D. Florescu, I. Manolescu, M. J. Carey, and R. Busse, The XML Benchmark Project, CWI (Centre for Mathematics and Computer

Science), Amsterdam, The Netherlands, 2001.

[SWK+01a] A. Schmidt, F. Waas, M. Kersten, D. Florescu, M. J. Carey, I. Manolescu, and R. Busse, Why and How to Benchmark XML Databases, ACM SIGMOD Record Volume 30, Issue 3, Pages 27-32, September 2001.

[WSII] Web Sphere Information Integrator (WSII), www.ibm.com/software/data/integration.

[WRS] WebRowSet XML Schema definition, java.sun.com/xml/ns/jdbc/webrowset.xsd.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Proxim-CBR: A Scaleable Grid Service Network for Mobile Decision Support

M. Ong, M. Alkarouri, G. Allan, V. Kadirkamanathan,
H.A. Thompson and P.J. Fleming

Rolls-Royce University Technology Centre in Control and Systems Engineering
Department of Automatic Control and Systems Engineering
The University of Sheffield

Abstract

With the emergence of Grid computing and service-oriented architectures, computing is becoming less confined to traditional computing platforms. Grid technology promises access to vast computing power and large data resources across geographically dispersed areas. This capability can be significantly enhanced by establishing support for small, mobile wireless devices to deliver Grid-enabled applications under demanding circumstances. Grid access from mobile devices is a promising area where users in remote environments can benefit from the compute and data resources usually unavailable while working out in the field. This is reflected in the aircraft maintenance scenario where decisions are increasingly pro-active in nature, requiring decision-makers to have up-to-date information and tools in order to avoid impending failures and reduce aircraft downtime. In this paper, an application recently developed for advanced aircraft health monitoring and diagnostics is presented. CBR technology for decision support is implemented in a distributed, scaleable manner on the Grid that can deliver increased value to remote users on mobile devices via secure web services.

1 Introduction

This paper describes a distributed, service-oriented Case-Based Reasoning (CBR) system built to provide knowledge-based decision support in order to improve the maintenance of Rolls-Royce gas turbine engines [1]. Figure 1 shows a typical gas turbine engine (GTE) used to power modern aircraft. As of 2006 there are around 54,000 of these engines in service, with around 10M flying hours per month. These GTEs, also known as aero engines, are extremely reliable machines and operational failures are rare. However, currently great effort is being put into reducing the number of in-flight engine shutdowns, aborted take-offs and flight delays through the use of advanced health monitoring technology. In this maintenance environment, decisions are increasingly pro-active in nature, requiring decision-makers to have up-to-date information and tools in order to avoid impending failures and reduce aircraft downtime. This represents a benefit to society through reduced delays, reduced anxiety and reduced cost of ownership of aircraft.

Grid computing [2] from mobile devices is a promising area where users in remote environments can benefit from the compute and data resources usually unavailable while working out in the field. This paper describes how Grid, CBR and mobile technologies are brought together to deliver a *substantial* further improvement in maintenance ability by facilitating the wide-scale communication of information, knowledge and advice between individual aircraft, airline repair and overhaul bases, world-wide data warehouses and the engine manufacturer. This paper focuses on *Proxim-CBR*, a recently developed system, as an example of how Grid technology can be extended to handheld

Figure 1. A Rolls-Royce Gas Turbine Engine

devices to support pro-active mobile computing in a dynamic aircraft maintenance environment.

2 Related Work

The *Rolls-Royce University Technology Centre* (UTC) *in Control and Systems Engineering* at The University of Sheffield has a long history of using CBR for engine health monitoring and diagnostics. The UTC has actively explored a variety of on-wing control system diagnostic techniques and also portable maintenance aid applications. CBR techniques were applied in developing a portable PC-based Flightline Maintenance Advisor [3, 4] to correlate and integrate fault indicators from the engine monitoring systems, Built-In Test Equipment (BITE) reports, maintenance data and dialog with maintenance personnel to allow troubleshooting of faults (figure 2). The primary advantage of the CBR-based tool over a traditional paper-based Fault Isolation Manual is its capability to use knowledge of the pattern of multiple fault symptoms to isolate complex faults. This effort was eventually escalated to the development of an improved Portable Maintenance Aid for the Boeing 777 aircraft. The outcome of these initiatives included the implementation of a portable Flightline Maintenance Advisor that was trialled with Singapore Airlines.

More recently, rather than using a portable computer which needs updating with new data as it becomes available, it was clearly desirable for a CBR system to be accessed remotely by engineers over a computer network. The advantage of this is that it is easier to support and it also allows search of an extensive knowledge base of historical maintenance incidents across an entire fleet of engines. This was evident in the outcome of a large-scale initiative that explored this potential and developed new solutions [5, 6]. The *Distributed Aircraft Maintenance Environment* (DAME) project was a pilot project supported under the United Kingdom e-Science research programme in Grid technologies. DAME was particularly focused on the notion of proof-of-concept, using the Globus Toolkit and other emerging Grid technologies to develop a demonstration system. This was known as the DAME Diagnostic and Prognostic Workbench that was deployed as a Web-enabled service, linking together global members of the DAME Virtual Organisation and providing an Internet-accessible portal to very large, distributed engine data resources, advanced engine diagnostic applications and supercomputer clusters that

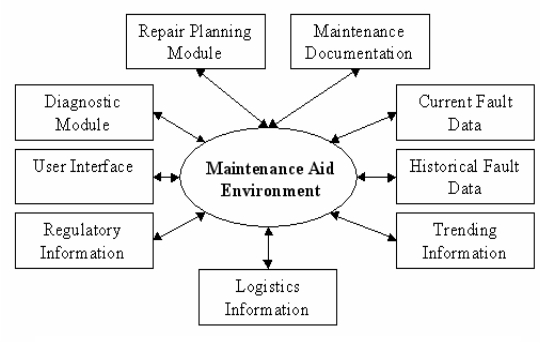


Figure 2. Structure of the Flightline Maintenance Advisor

delivered the required processing power. DAME tackled complex issues such as security and management of distributed and non-homogenous data repositories within a diagnostic analysis framework with distributed users and computing resources.

Current research includes work being done in the *Business Resource Optimisation for Aftermarket and Design on Engineering Networks* (BROADEN) project. BROADEN is a UK DTI funded project [7], led by Rolls-Royce Plc, that aims to build an internal Grid at Rolls-Royce to further prove, in a production environment, the technologies developed in DAME. This will include integrated diagnostic and prognostic tools for health monitoring across development test, production pass-off and in-service aero engines, and the formulation of a strategy to transfer proven Grid technology to production networks. This project supports the "Health Management and Prognostics of Complex Systems" research theme from the Aerospace Innovation and Growth Team's National Aerospace Technology.

3 Engine Diagnostics and Maintenance

A fundamental change of emphasis is taking place within companies such as Rolls-Royce Plc where instead of selling engines to customers, there is a shift to adoption of power-by-the-hour contracts. In these contracts, the engines are effectively provisioned to clients whilst the engine manufacturer retains total responsibility for maintaining the engine. To support this new approach, improvements in in-flight monitoring of engines are being introduced with the collection of much more detailed data on the operation of the engine. New engine monitoring equipment [8] can record up to 1GB of data per engine for each flight. In the future, Terabytes of data could be recorded every day for diagnostic and prognostic purposes.

3.1 The Data Capture Process

The Engine Control Unit (ECU), also known as a Full Authority Digital Engine Controller (FADEC), has a test port to which the Engine Management System (EMS) connects to access data. The ECU test point can provide many data parameters. Other systems can interface with the EMS and gain access to any of the data provided by the ECU. The Engine Monitoring Unit (EMU), as fitted to newer engines, includes sophisticated built-in vibration monitoring functionality and specialized vibration feature detectors. A Digital Flight Data Recorder (DFDR) records the aircraft airframe and engine parameters and can be downloaded after every flight. At present, snapshots of key engine performance data at automatic intervals are sent to ground-based systems via the Aircraft Communications Addressing and Reporting System (ACARS), along with routine data such as departure reports, arrival reports, passenger loads, fuel data, and much more. This data is transmitted by VHF radio link and satellite. However, the costly pay-per-kilobit ACARS is deemed unsuitable for moving large quantities of data. In the near future, large quantities of engine performance and flight data can be automatically transferred, via Gatelink [9] at airports, to the ground-based Grid system for routine processing [10]. Gatelink is a cost effective, short-range, spread spectrum, broad-band, wireless, microwave datalink that can rapidly move large quantities of data on and off the aircraft.

3.2 The Diagnostic and Maintenance Process

Diagnosis is defined as the act or process of identifying or determining the nature and cause of a fault through the evaluation of symptoms, history and examination. Prognosis is a prediction of a probable course of an emerging fault through evaluation of early symptoms.

A typical scenario is as follows: The on-wing diagnostic system and its associated ground based system are used prior to the use of the Grid-based system. In addition to that initial on-wing diagnosis, the Grid-based system is always used to provide an automated diagnosis. This is desirable because it can detect additional situations such as a recurring errant diagnosis, a new condition that has not yet been uploaded to the on-wing monitoring system, or a condition that can only be detected using tools that require extensive ground-based processing facilities. The resultant automatic diagnoses can then be assessed. In the vast majority of cases normal situations are indicated, however, if a condition is detected with a known cause then appropriate maintenance action can be planned. Additionally, in the rare case that a condition is detected without a clear cause then the situation will be escalated, within the system, to one of various remote experts who can look into the matter further. Using the Grid-based system, they will have access to the data from the current engine flight. They are able to run searches on historical data, get knowledge-based maintenance advice, run signal processing and run simulation tasks to gain an insight into any given event.

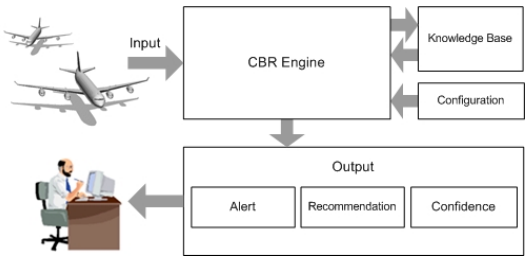


Figure 3. CBR System Architecture

4 CBR for Decision Support

Case-Based Reasoning (CBR) is a knowledge-based, problem-solving paradigm that resolves new problems by adapting the solutions used to solve problems of a similar nature in the past [11, 12]. A further advantage of this approach is that it allows consolidation of rule knowledge and provides a reasoning engine that is capable of probabilistic-based matching. With CBR technology, development can take place in an incremental fashion facilitating rapid prototyping of an initial system. The development of robust strategies for integration of multiple health information sources is achieved using reasoning algorithms of progressively increasing complexity.

In contrast to conventional search engines, CBR systems contain a knowledge model of the application domain in which it operates on. It is therefore not universal but specifically designed for the domain. Hence, it is possible to develop intelligent search abilities, which even show reasonable results when given fuzzy or incomplete requests. Moreover, the results are ranked and complemented by variants and alternatives, thus, not only matches are given but information is valued with "more suitable" or "less suitable". Figure 3 depicts the high-level stages of reasoning

in our implementation. However, this does not reflect the actual software architecture. It depicts the functional as opposed to physical configuration. In the following sections, these components are described in further detail.

4.1 Knowledge Base

Essential to our CBR system is the casebase that represents the knowledge repository. This contains detailed descriptions of engine faults and the best practice maintenance advice (solutions) gathered from engineers and experienced mechanics over the development and service life of engines. For a new engine type, little information is known initially but with our CBR technique, a casebase of independent records of fault and maintenance information can be developed in a piecemeal manner and updated as and when knowledge about the behaviour of the engine is known. More importantly, the location of the knowledge base within a virtual maintenance facility on the Grid also allows the integration of diagnostic knowledge from multiple health information sources which are vital in improving the accuracy and coverage of the CBR system. Useful diagnostic information previously available from separate monitoring and maintenance systems, when brought together into a single system, provides for a more powerful diagnostic tool.

4.2 CBR Engine

The primary responsibility of the CBR Engine is to read the knowledge base into memory and perform retrieval, matching and ranking of the cases based on specific query input. The CBR Engine also provides the interface for the system to generate new cases, view and manage existing cases, manage the knowledge model and execute any external modules that contribute to the analysis of information within the system. Retrieval, matching and ranking are performed using an enhanced, weighted nearest-neighbour algorithm of progressive complexity. Using a well defined, internal Knowledge Model of the domain that it operates on, the algorithm effectively emulates how an expert would identify the problem from past knowledge by performing similarity measures across the available information. Although this is done automatically, trained users are allowed to influence the entire process by choosing specific attributes to analyse. They can configure specific weighting of attributes to correspond to the "importance" of each chosen attribute. In addition to that, hard constraints can be defined prior to that so the algorithm can optimise the entire process by reducing the number of potential cases to be matched. With a continuously expanding knowledge base of cases at a global scale, the process described above presents an ever increasing demand for computing resources.

4.3 Recommendation

Traditionally, existing cases in the knowledge base are adapted to form a solution for a new problem. In this diagnostics and maintenance environment, however, the output of the CBR system is delivered in the form of a discrete set of potential candidates as opposed to a single solution. Concern for accountability in a safety-critical environment often dictates that the system not force a user into accepting a single solution. The system aims to recommend a narrowed-down choice of possible solutions, using knowledge accumulated previously about the domain on which it operates. This will enable the user, an aircraft expert in particular, to make an informed decision. In order to do that, each solution is accompanied by the system's confidence in that answer. Various threshold levels can be tuned to both limit the number of potential solutions as well as eliminate inadequate confidence. Ultimately, the system acts to support the decision making task, but the aircraft expert retains the responsibility for the final decision.

4.4 Confidence

As described above, confidence is one of the critical factors of the CBR system in recommending solutions to the expert. The algorithm will compute the confidence of a solution by taking into account prior knowledge contained in the knowledge model, which includes a model for natural-language terminology, boundary values for each attribute, and attribute relationship models. The process is repeated for each candidate solution to finally obtain a ranked list of recommended solutions. It is important to note here again that a trained user is able to fine-tune the entire process by selecting specific attributes, configuring the weighting of each attribute to correspond to the relative "importance" of that attribute, and also define hard constraints so that the algorithm can further optimise the process.

5 Grid Computing and CBR Deployment

The Grid is defined as an aggregation of geographically dispersed computing, storage and networking resources [2]. It provides many advantages over conventional high-performance computing because it is co-ordinated to deliver improved performance, better utilisation, scalability, higher quality of service and easier access to data. This makes the Grid ideal for collaborative decision support across virtual organisations because it enables the sharing of applications and data in an open, heterogeneous environment. Applications previously considered to be host-centric, such as the CBR system described in the previous sections, can now be distributed throughout a Grid computing system, thus improving quality of service while also offering significantly enhanced capabilities. Our use of industry standard technologies to implement such a Grid-enabled CBR system is discussed here.

5.1 Grid Service

Grid Services are essentially Web Services with improved characteristics such as state and life-cycle management [13]. Our CBR system's functionality has been delivered as a Grid service in a Service-Oriented Architecture (SOA) environment. SOAs are essentially a collection of services, focusing on interoperability and location transparency. These services communicate with each other, and can be seen as unique tools performing different parts of a complex task. Communication can involve either simple data passing or it could involve two or more services co-ordinating some activity. SOAs and services are about designing and building a system using heterogeneous network addressable software components. An important aspect of the service-based CBR delivery is that it separates the service's implementation from its interface.

A typical use scenario for the CBR service is as follows: A Web browser initiates a request to the CBR service at a given Internet address using the SOAP (Simple Object Access Protocol) protocol over HTTP (Hypertext Transfer Protocol). The CBR service receives the request, processes it, and returns a response. Based on emerging standards such as XML (eXtensible Markup Language), SOAP, UDDI (Universal Description, Discovery and Integration) and WSDL (Web Service Definition Language), the CBR service along with other diagnostic services form a distributed environment in which applications, or application components, can inter-operate seamlessly across the virtual organization in a platform-neutral, language-neutral fashion. Consumers of the CBR service are allowed to view the service simply as an endpoint that supports a particular request format or contract. Consumers need not be concerned about how the CBR service goes about executing their requests; they expect only that it will.

5.2 High Performance Computing

High performance computing support for the CBR service is provided by means of implementing Open Grid Services Architecture (OGSA) concepts with Web Service technologies. With this, the above-mentioned CBR service has benefited from both Web Service technologies as well as Grid functionality. More specifically, the compute-intensive tasks within the CBR matching process, previously computed on a single computer, can now be aggregated across any available Grid supercomputing node with the use of Grid middleware, the Grid's effective operating system [14]. With this, multiple instances of the CBR service can be generated on-demand. The advantage of this is that it allows multiple consumer requests to be processed simultaneously by separate nodes of the Grid. With our CBR system as a prime example, the integration of Grid computing with Web services has benefited aircraft experts at any remote geographical location by providing them with access to powerful diagnostic tools, data repositories and large computing resources via any Internet-enabled computing device.

6 Grid-Enabled Decision Support on Mobile Handheld Devices

In addition to the typical diagnostic and maintenance scenario described previously, a situation may arise where an aircraft engine expert, usually regarded as a high-value resource, is required to investigate a currently occurring problem under demanding circumstances; i.e. he/she may be traveling to a foreign aircraft site and a conventional computer is not available. Furthermore, the ability of a handheld computer to capture investigative findings on-site and instantly upload these findings to the Grid-based system greatly increases the accuracy of the CBR system in diagnosing the problem. The advantage in

rapidly sharing the information across the entire system is that it enables several other engine experts, at different geographical locations, to collaborate in solving the problem, thus reducing aircraft down-time and costs. In the following sections, we discuss the difficulties faced with currently-available mobile, handheld computers and our efforts in developing solutions to these problems.

6.1 Limitations on a Typical Device

A mobile computing device such as the widely available Personal Digital Assistant (PDA) shown in figure 4, presents various problems such as hardware constraints, small graphical display area and security issues. The limited processing power and built-in memory means that applications need to be efficiently implemented. In practice, this will limit the complexity of useable application programming interfaces (API). Software development for a mobile, handheld device tends to be more complicated to account for the device's operating system and individual hardware platform. Furthermore, a small display area usually found on such devices can also make presentation of information to the users a challenging task.

6.2 PDA Demonstrator

In an effort to overcome these limitations, a demonstrator consisting of a PDA that utilises the previously mentioned Grid-based CBR service was designed and implemented. The primary aim of this system was to aid diagnostic and maintenance of gas turbine engines from remote locations under demanding circumstances. The mobile, handheld computing device used in this work is a widely available *HP- iPAQ* PDA. It features, as standard, a built-in 802.11b [15] wireless network adaptor, Bluetooth [16, 17] connectivity and a mini Web browser. Using these, the PDA can to connect to the Internet via a wireless LAN (Local Area Network) or via Bluetooth to a suitable GSM (Global System for Mobile Communications) [18, 19] cellular phone. For the latter option, data transfer using a GPRS (General Packet Radio Service) system [20] on a GSM network is preferred over regular GSM because of the higher bandwidth it provides. This is used to access CBR services available on the Grid. The real advantage of this is that it can offer a user all the benefits of large, global data and high performance applications required in the diagnostic and maintenance scenario.

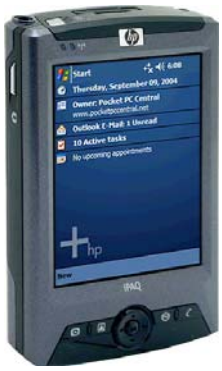


Figure 4. A Typical mobile handheld PDA

The limited capacity of PDAs available currently make it impossible to have a complete Globus toolkit equivalent implemented on the device. This necessitates the use of a host-side proxy that will interact with the Grid environment whilst providing suitable access for the device. Here, a mini Web portal, implemented with Apache Tomcat, is used as the proxy, enabling access for the device to Grid services using a standard, built-in mini Web browser that acts as the client to the proxy. The mini Web browser represents the front-end for the user on the PDA.

The mini portal is very similar to a standard portal accessed by conventional desktop computers, the primary difference being that the mini-portal is simplified by removing complex script functions normally aimed to enhance the layout of content on larger displays. Furthermore, the page layouts are rearranged such that each page would only execute a single CBR operation at a time, both for display purposes and to optimise on bandwidth usage. In an evaluation of

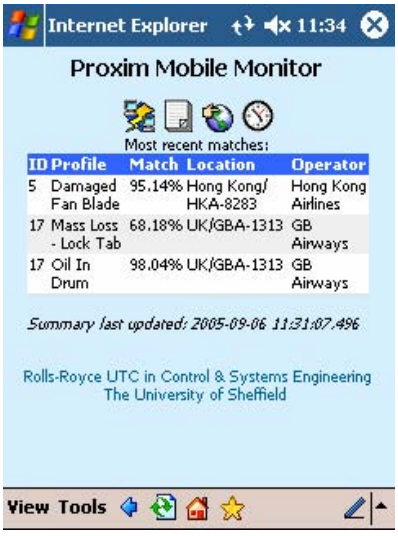


Figure 5. Fault alert and diagnosis

the mini portal against a standard web portal, it has been found that the mini portal can deliver an identical quality of results to the user whilst minimising the graphical content load on the PDA display. An Apache Tomcat servlet container serves as the hosting environment for the mini portal on any Grid node.

The software and communication standards used in the demonstrator systems that have been implemented are based on widely accepted industry standards thus it is highly feasible for the system to be migrated across various other application areas. Furthermore, the increasing availability of wireless networks [17, 21] and advances in Grid technology provide a strong case for mobile, Grid-enabled decision support with handheld computing devices.

In the following sections, the CBR system, Grid computing and mobile PDA work described thus far are implemented in two different areas of the aero engine diagnostic and maintenance scenarios – pro-active remote condition monitoring and fault investigation.

6.3 Pro-Active Remote Condition Monitoring

Condition monitoring systems are synonymous with fault diagnostics. Whether localised or distributed, they are common and widespread, with more and more organisations becoming dependant on such systems to support critical decision making in order to identify problems quickly, avoid unnecessary risks and reduce cost.

In the past, centralized high-performance ground-based systems have the ability to support powerful applications but they are limited in their scope of reach or can prove inaccessible when and where they could be most effective. At the opposite end of the scale, portable or on-board standalone systems can be very useful in remote locations or harsh operating environments but they are very limited in terms of performance and capacity.

Today, the integration of service-oriented architectures, modern wireless technologies, advanced mobile devices and distributed computing presents a unique solution to overcome these limitations. This integration can provide the necessary high-performance computing capability required to continuously perform condition monitoring whilst transmitting essential information back to key users via a mobile handheld device. Figure 5 depicts a screen capture of this on a PDA that was implemented as part of our work. Here, possible causes of the occurring problems on different aircraft have been successfully identified. Note that any references to actual aircraft or operators have been anonymised.

The prime advantage of this approach is that any occurring problem condition can be made known immediately to the right person, in the right place at the right time, giving that person the opportunity to pro-actively handle the arising problem condition and collaborate on-line with a virtual network of experts to solve the problem. Complementary analysis tools and services on the Grid allow a widespread collection of knowledge, data and tools that were not previously available in this manner to be used on-line for further investigation. It is this model of problem-solving that we define here as pro-active mobile computing for decision support.

The condition monitoring demonstrator system has been implemented using a network of remote CBR service nodes deployed at different geographical locations. This will be described in close detail in Section 7. These remote nodes continuously receive diagnostic information, downloaded from aircraft on-wing systems, from multiple sources in real-time. Any newly received information will be automatically analysed to identify problems.

A unique feature of our distributed architecture is the ability to monitor health information across multiple nodes, at different locations, without actually transmitting that information off-site. In the event of a problem, alert messages can be automatically flagged to notify remote users via suitable communication channels based on the severity of the problem. At this point, the actual exchange of information between stakeholders can then be negotiated via established, trusted channels for further action based on the organisations' requirements and service-level agreements. A clear advantage of this approach is the significant reduction in the network bandwidth required. But more importantly, it further supports crucial security policies and privacy of any commercially sensitive data.

6.4 Pro-Active Fault Investigation

For a more user-driven, interactive fault investigation process, the PDA-based mini browser can be used to transmit specific user inputs to the CBR service in order to match the details of a currently occuring problem against the large CBR knowledge base on the Grid. Figure 6 depicts the match results of such a process on the PDA. For any new or existing case, additional knowledge gained from further analysis such as vibration spectral data, as depicted in figure 7,

digital images or even audio/video media can be appended to the case in real-time as it is being investigated. This makes the information available for immediate use by other experts across the Grid and if necessary, escalated to another expert at a different location for futher investigation. Finally, problems that have been identified and the successful solution to that problem can be appended to the case, on-line, via the same interface. This will be stored in the CBR knowledge base on the Grid as new knowledge for future use (figure 8).

To further support the diagnostic process, engine performance simulations, normally a compute-intensive and time consuming task, can be executed using an Engine Simulation Grid Service (ESGS) [22]. The results of this can then be retrieved on the mobile device. This facility has been made available on our PDA demonstrator system, as presented in figure 9. Depicted in this figure is crucial information relating to a particular event, which was later identified by engine experts as a situation where an abnormal spike occurred in the engine data due to a bird-strike. In this situation, a large bird is accidentally ingested by the engine whilst in flight. Such an occurrence usually goes unnoticed by the pilot or aircraft operator, but can potentially damage to the engine's internal components. With this available knowledge, a remote expert may decide to monitor the engine more closely and arrange for a boroscope inspection of the engine's internals at the next convenient landing location.

7 Proxim-CBR Grid Service Network

In order to support the pro-active diagnostic and maintenance process, the Grid-based CBR system is made up of a scaleable network of CBR services. Three different types of CBR nodes exist within the system. Each node is deployed as a Grid service on a remote host computer. Figure 10 depicts these node types and their relationship in the node network. Their details are described in the following sections.

7.1 Proxim Service Node

The Proxim Service Node, as depicted in figure 10, is located at the back-end of the node hierarchy. This node is typically deployed at the geographical location where data is first captured in the system. It contains a CBR Engine and hosts a repository of cases local to this node. When requested by a Proxim Broker Service Node, this

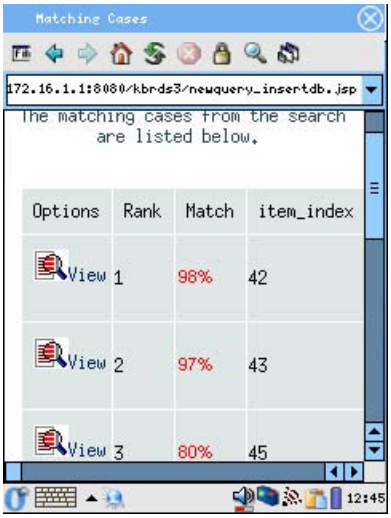


Figure 6. Pro-active fault investigation

node will perform a search, match and rank process at a local level. It then returns a summary of results to the requesting broker. This node is also responsible for delivering complete case records to a human user via the broker. However, the latter function is subject to the user being granted appropriate access rights to that complete record.

7.2 Proxim Broker Service Node

The Proxim Broker Service Node maintains a comprehensive registry of all available Proxim Service Nodes. When requested by a Proxy Service Node, it will initialize and orchestrate a search, match and rank process across all the available Proxim Service Nodes. Individual sets of match results received from every Proxim Service Node are collated to form a single result summary. A summary will also include essential information needed to retrieve an individual result case in full detail when required. For interactive, fault investigation purposes, queries can be submitted to the Proxim Broker Service Node directly from the CBR portal on the Internet. For automated condition monitoring purposes, a Proxy Service Node is deployed to automatically initialize and manage multiple queries to the Proxim Broker Service Node. When any query is submitted, the CBR system effectively views the distributed collection of remote data as a single, virtual repository.

7.3 Proxy Service Node

A Proxy Service Node contains a repository of query profiles. Each profile represents a known problem condition or engine event description. It also maintains a registry of available Proxim Broker Services. With these in place, the node will automatically initialize and manage query processes to an available broker to identify problem conditions. Depending on the requirements of the application, this can be configured to execute at automatic intervals or in a continuous manner. Result summaries generated by the Proxim Broker Service are stored in the Proxim Mobile Monitor Service. This will be presented via a PDA to a human user, in particular an aero engine expert, whenever and wherever required. It is important to note here that there may exist more than one Proxy Service Node, each one will support queries to single or multiple brokers at different locations. When deploying multiple Proxy Service Nodes, the functionality to be supported by each node is determined by the needs of the intended application domain or by specific, unique areas of a particular system being monitored. For example, there could be many proxies operating on different components of an aircraft at different intervals. In another example, a proxy could be deployed specifically for, and wholly owned by, an individual or organisation that has query profiles representing knowledge of some commercial value.

8 Security

The dynamic and multi-institutional nature of Grid environments introduces challenging security issues that demand new technical approaches [23]. For instance, the CBR knowledge base and related engine diagnostic data could contain highly relevant information about an engine's design characteristics and operating parameters. This could potentially be misused. For this reason, access to the systems described here is highly restricted to authorised users only. To overcome the problem, the Grid Security Infrastructure (GSI) provided by the Globus Toolkit has been used to enable secure authentication and secure communication over an open network. GSI is composed of security elements that conform to the Generic Security Service API (GSS-API), which is a standard API for security systems promoted by the Internet Engineering Task Force (IETF).

GSI consists of a number of security features that include mutual authentication and single sign-on. This is based on public key encryption, digital certificates and Secure Sockets Layer (SSL) communication. At the core of the security

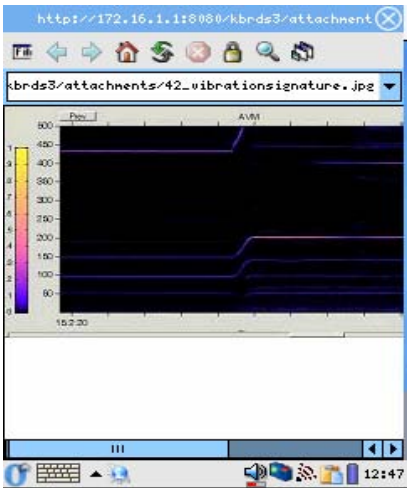


Figure 7. Engine vibration spectral data

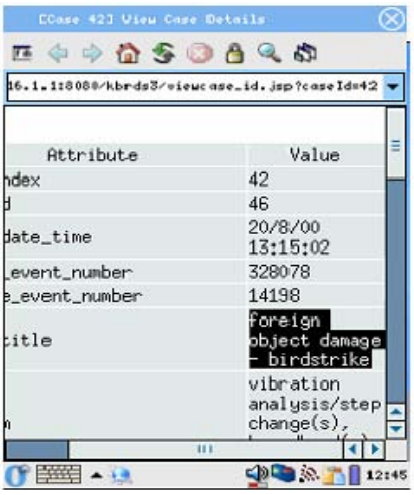


Figure 8. Full fault case details

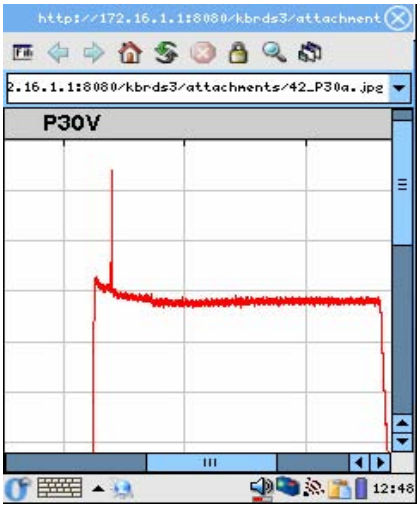


Figure 9. Abnormal spike in engine data identified via the Engine Simulation Grid Service

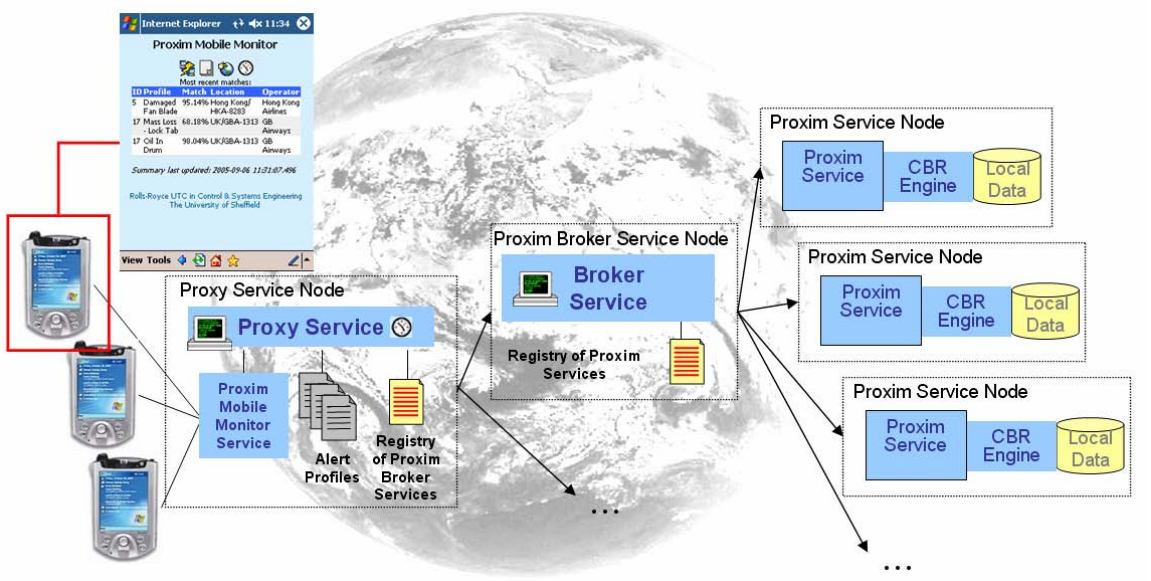


Figure 10. Scaleable Proxim-CBR Grid Service Network

infrastructure is authorisation based on X.509 digital certificates for both service consumers and service hosts. Hence, all users and service hosts need to acquire a certificate issued by a trusted Certificate Authority (CA). The bottom line is that any user, regardless of his/her physical location, will be barred access to any resource on the system unless their credentials have been successfully verified.

9 Conclusions and Future Directions

The capability displayed by the PDA demonstrator system is particularly important because it offers aircraft experts, considered as a high-value resource, the mobility to pro-actively operate on large data and complex problems in an “anytime and anywhere” manner. With CBR technology, problem identification in health monitoring is not limited by a set of rules, but the overall "closeness of match" and ranking of a problem condition can be identified. In some situations, this may be used to prompt an investigation to facilitate early detection or on-condition monitoring. In the longer term, CBR will contain a constantly growing repository of event knowledge that represents a very valuable knowledge asset for the domain operator.

Similar scenarios commonly occur across a diverse range of domains and this includes engineering, healthcare and finance. However, regardless of the application area, each system shares a number of similar operating and design characteristics, making it possible to experience the similar benefits of Grid-enabled, pro-active mobile decision support. One such area is large-scale monitoring of patient health. Using Proxim-CBR, a medical expert could potentially have the capability to continuously monitor the condition of multiple patients at different geographical locations whilst maintaining the privacy of their medical records.

Our work in this paper has demonstrated that Grid-enabled computing from mobile devices can be very effective in a decision support environment. The real advantage of service-oriented architectures is that it enables business/engineering processes to be examined so that services can be provided to 'accurately and sufficiently' support core operations. In the past, there has been a tendency for organisations with complex activities to be burdened with unnecessary bureaucracy and/or insufficient timely information/detail. If engineered correctly, scaleable Grid service networks, such as Proxim, have the potential to liberate enterprises into more effective and dynamic management of their processes, and they no longer have to force-fit a "one size fits all" solution to their problem. Each architecture solution can be specific, and this is where pro-active mobile computing can be particularly powerful, i.e. in some situations, say in bedside medical treatment, it has not been possible to have powerful diagnostic support tools and information, despite the fact that is where indeed the support/tools are most needed. The real

achievement of the Proxim-CBR PDA condition monitoring demonstrator presented here is that there is now the opportunity to properly support processes that have always needed mobile decision and information support. The work presented in this paper extends the range of process support to where it is most needed, i.e. where the work is being done.

Acknowledgements

This project is part-funded by a Collaborative R&D grant under the DTI Technology Programme. Further information can be found at www.dti.gov.uk/technologyprogramme. The authors gratefully acknowledge contributions from Rolls-Royce plc and partners of the BROADEN project.

References

[1] Rolls-Royce. "The Jet Engine", Rolls-Royce Plc, 1986.
[2] I. Foster (Ed.) and C. Kesselman (Ed.), "The Grid 2", Morgan Kaufmann, 2003.
[3] S.M. Hargrave. "Evaluation of Trent 800 Portable Maintenance Aid Demonstrator", Rolls-Royce University Technology Centre, University of Sheffield, Report No. RRUTC/Shef/R/98202, 1998.
[4] S.M. Hargrave. "Review of Performance-Based Diagnostic Tool", Rolls-Royce University Technology Centre, University of Sheffield, Report No. RRUTC/Shef/TN/98204, 1998.
[5] Distributed Aircraft Maintenance Environment (DAME) project; www.cs.york.ac.uk/dame, 2003.
[6] T. Jackson, J. Austin, M. Fletcher, and M. Jessop. "Delivering a Grid enabled Distributed Aircraft Maintenance Environment (DAME)", Proc UK e-Science All-Hands Meeting, pp. 420-427, 2003.
[7] Business Resource Optimisation for Aftermarket and Design on Engineering Networks (BROADEN) project; www.shef.ac.uk/acse/research/themes/utc/broaden.html, 2005.
[8] G.F. Tanner and J.A. Crawford. "An Integrated Engine Health Monitoring System for Gas Turbine Aero-Engines", Proc. IEE Seminar on Aircraft Airborne Condition Monitoring, pp 5/1-5/12, 2003.
[9] J. Croft. "Avionics: Beaming Bits and Bytes", Air Transport World, p. 54, January 2006.
[10] H.A. Thompson, "The Use of Wireless and Internet Communications for Gas Turbine Engine Monitoring and Control", Proc AIAA/ICAS International Air and Space Symposium and Exposition: The Next 100 Years, July 2003.
[11] J. Kolodner. "Case-Based Reasoning", Morgan Kaufmann, 1993.
[12] S.K. Pal, T.S. Dhillon and D.S. Yeung (Eds). "Soft Computing in Case Based Reasoning", Springer-Verlag UK, 2000.
[13] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. "Grid Services for Distributed System Integration", Computer, 35(6), 2002.
[14] I. Foster and C. Kesselman. "Globus: A Metacomputing Infrastructure Toolkit", Intl J. Supercomputer Applications, vol. 11, no. 2, pp. 115-128, 1997.
[15] IEEE 802.11 Working Group, "The IEEE Std 802.11b-1999", 1999.
[16] IEEE 802.15 Working Group. "The IEEE Std 802.15.1-2002", 2002
[17] A. Vollmer. "With Devices Ready to Go, Bluetooth is Poised to Make Its Move", Electronic Design, July 24, 2000.
[18] M. Mouly and M. Pautet. "The GSM System for Mobile Communications", Published by the authors, 1992.
[19] M. Rahnema. "Overview of the GSM System and Protocol Architecture", IEEE Communications Magazine, vol. 31, no. 4, pp. 92-100, April 1993.
[20] R. Kalden, I. Meirick, and M. Meyer, "Wireless Internet Access Based on GPRS," IEEE Personal Communications Magazine, vol. 7, no. 2, pp. 8-18, April 2000.
[21] U. Varshney. "Recent Advances in Wireless Networking", IEEE Computer, vol. 33, no. 6, June 2000.
[22] X. Ren, M. Ong, G. Allan, V. Kadirkamanathan, H.A. Thompson and P.J. Fleming. "Service-Oriented Architecture on the Grid for Integrated Fault Diagnostics", Journal of Concurrency and Computation: Practice and Experience, John Wiley and Sons, 2006.
[23] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. "Security for Grid Services", Proc. 12th IEEE Int'l Symposium on High Performance Distributed Computing (HPDC-12), pp. 48-57, IEEE Press, 2003.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Grid monitoring: a holistic approach.

A.P. Millar[1]

1 - Dept. of Physics and Astronomy, University of Glasgow, Glasgow, G12 8QQ.

Abstract

Computational grids involve the intersection of different geographically distributed communities: the resource-users and the resource-providers. Grid monitoring is required by various super-sets of these communities as everyone wants to know something about how the Grid is performing.

Many grid monitoring systems have a top-down prescriptive approach. Grid monitoring is often considered separable from the monitoring of non-grid elements within a functioning grid site. This schism can result in grid monitoring ignoring existing technologies, often requiring some form of bridging between different systems.

In this paper, the key interested parties of grid monitoring are discussed along with what information they are interested in. The three-component model of monitoring systems is presented and illustrated with examples. From this model, we deduce the concept of a universal sensor and discuss its role in providing different information to different people.

The `MonAMI` project is introduced with its aim of providing an implementation of a universal sensor. Possible uses of `MonAMI` are discussed.

1 Introduction

A Grid provides computational resources that many end-users would like to use. Like any computational resource, Grid resources can suffer from performance problems or even complete outages. Sometimes end-users can precipitate problems by following unexpected usage patterns. These unusual usages can cause greatly reduced performance or may even cause services to fail. Under these circumstances it is important to establish what triggered a problem so it can be prevented in the future.

None of these requirements are unusual in computational services. What makes monitoring in a grid environment challenging is the interaction of the geographically distributed groups of people.

1.1 Who wants to know?

Consider a (non-Grid) service that attracts many users. These users may need to know the current status of the service. To provide this information, the service provider can monitor the service and make current status information available through a centralised set of webpages. Often the service is distributed, but will have sufficient cohesion to enforce a single monitoring solution for the different components.

One possible translation of the central monitoring to Grid-based computing is that a single site might provide information on how their components are performing. This information would give an indication of how heavily their resources are being used. This information would be gathered (from available sensors), collected and presented perhaps on a suitable set of web pages. The gathered information would also alert the site administrator when a service is failing or performing badly, allowing a rapid response in fixing the problem. Site administrators are principally interested in whether they are providing a working service, that the resources (e.g. free disk space) are sufficient for medium term projected usage and whether they are satisfying their agreed service provision.

Another translation of this central monitoring is to a Virtual Organisation (VO). A VO represents a group of Grid users with similar aims and access to similar computational resources. They might want to monitor the Grid with particular views specific to the tasks they wish to undertake. Each VO can undertake monitoring of resources (using available sensors) and provide customised views of available data. The VO may have specific software or site-local service requirements, the availability of which dictates which sites they can use. They might also wish to conduct more detailed monitoring of the services they use to check for bottle-necks, both from their code-base and from the available sites.

A third group interested in monitoring are the software developers. With small-scale deployments, it is possible for software engineers to gain access to servers to look for the cause of performance problems. With wide-scale production deployment, this is no longer feasible. Instead, the monitoring infrastructure must allow people to gather more detailed information. This information is only needed when a bottleneck is discovered. Collecting this information routinely would be prohibitive, so the monitoring system must allow monitoring on-demand.

Yet a fourth translation of centralised monitoring is for "the grid" to monitor itself. Any grid will provide grid-level functionality: activity based on multiple sites providing similar services. Often, these services can be tested in isolation (as part of the site-level tests), but a much broader testing can only be conducted by using the multi-site functionality. Centralised grid-level monitoring tests the multi-site functionality. It is similar to the site-level monitoring but includes tests that site-administrators cannot conduct.

To illustrate introspective monitoring (the ability of a grid to monitor itself), the grid established for the forthcoming Large Hadron Collider (LHC) facility (based at CERN) will be examined briefly. The expected rates of data coming from the various LHC experiment detectors is immense: in excess of 14 PB per year (or 135 TB per day). To support processing this volume of data, the CERN partner countries have established a grid called the World-wide LHC Computation Grid (WLCG) (for further details see [1]).

Within the WLCG, sites are not truly autonomous but must go through a vetting process before becoming part of the grid[2]. Within the UK, contribution to this grid effort has been coordinated through the GridPP project[3]. Membership of these groups engenders a sense of community between the collaborating sites, moreover membership of WLCG is subject to signing a Memorandum of Understanding (WLCG MoU[4]) stating the expected level of service provision. Under these circumstances centralised Grid-wide monitoring is strongly required. Various centralised monitoring projects exist within WLCG[5] and GridPP[6].

In summary, a Grid service running on a particular site might be monitored by the local site-administrator, each of the different VOs that use the site, by the different software developers and centrally across the grid. The monitoring requirements are not static. Over time, other groups of users might require additional monitoring if, for example, new VOs are formed.

Each group interested in monitoring grid services might use different systems for monitoring as there is currently no single well-accepted, universally deployed monitoring system. This lack of consensus presents a problem in gathering the information needed to providing the monitoring information.

One possible solution to this problem is presented in this paper. The work is open-ended and is suitable for collaboration.

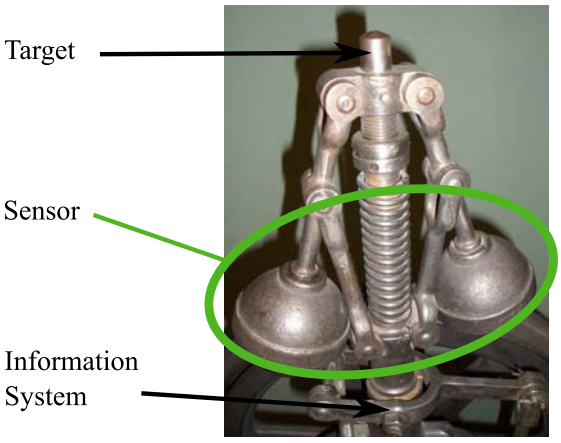


Figure 1: *An example of the three-component model implemented in hardware.*

2 Terminology

In discussing monitoring systems, it is useful to define the terminology that will be used throughout this paper.

Different monitoring systems exist with varying complexity. These systems may have many different components with which they provide the monitoring service. However, at the most abstract level, all monitoring systems can be understood in terms of the three-component model.

2.1 The three-component model

The three-component model places components of a monitoring system into one of three categories: target, sensor or information system.

A target is the object of the monitoring, something one wishes to ascertain its current state. It is a common feature amongst targets that, although they might provide some facility by which their current state can be monitored, they do not actively undertake any monitoring themselves. Examples of targets are file-systems, databases, or grid services.

The information system provides some method of storing the target's current state. An information system has no knowledge of how to get information from the target; it only provides a medium to store the captured information. Examples of information systems include a webpage, some portion of memory, a database, or an email delivered to the site administrator.

A sensor is a component that is sensitive to the target's current state. It uses this sensitivity to gather the target's current state and store this data within some information system. A sensor translates information from the target to the information.

A simple mechanical example of the three-

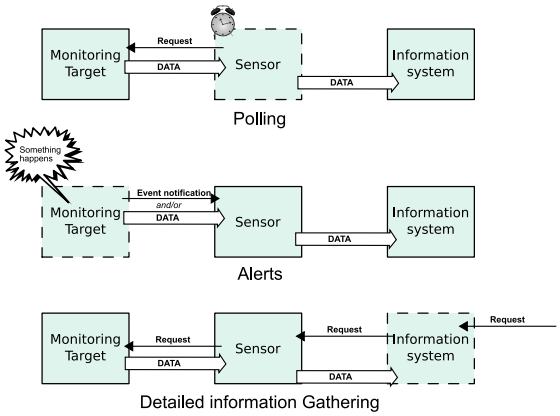


Figure 2: *The three monitoring dataflow models: polling, alerts and detailed information gathering (DIG). The dashed box indicates which component initiated the request for information.*

component model (the Watt governor) is shown in Figure 1. Here, the monitoring target is the engine, the sensor is the pendulum (which is sensitive to the rotational speed) and the information system is the height of the collar.

2.2 The three types of dataflow models

Monitoring involves the flow of information from the target to the information system. Using the three-component model, we can classify these interactions as one of three types, based on which component initiated the flow of information.

The three types of monitoring interaction are polling, alerts and detailed information gathering (DIG). These correspond to the dataflow being triggered by the sensor, target and information system respectively, as shown in Figure 2.

Support for polling dataflow is perhaps the most commonly implemented system. The sensor, acting on an interrupt from an internal timer, reads the current state of the sensor. It stores this information within the information system. This information is typically displayed as a graph or subject to further analysis.

Support for alerts dataflow is available from some targets. Typically, this dataflow is triggered by some device or software service and will alert the sensor of some asynchronous event. Examples of asynchronous events include user activity or running out of some resource. The sensor can translated this information into a suitable form before sending it to the information system. This can be used for accounting, alerts when some component is failing or logging activity.

The third dataflow model is DIG. Some external agent (e.g. the end-user) requests additional information. The request for detailed information is processed and the results are sent back to the information system. One possible use for DIG is when a developer or end-user is investigating a problem with a production service. Another example of DIG is gathering configuration parameters to check that services have been configured correctly.

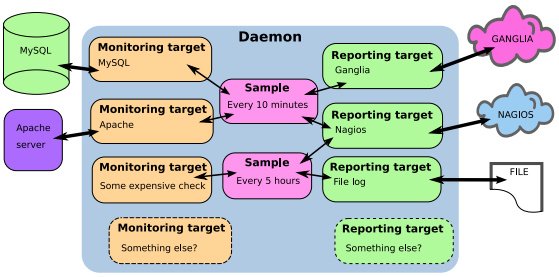


Figure 3: *Components of* `MonAMI`*, illustrated with a typical configuration for site-local monitoring.*

2.3 Hybrid dataflow systems

A three-component system, although useful in describing the basic monitoring interactions, is often not found in real-world implementations; instead, interactions between different components give rise to more complex behaviour. In general, complex systems can be broken down into component parts, each of which follow the three-component model.

For example, a monitoring system might watch performance metrics and send an email when the values pass beyond acceptable limits. This system can be broken down into two subsystems: one that periodically monitors the current performance, the other that sends an email when there is a problem. The first subsystem operates under the polling dataflow model, periodically updating the current measured values. The second subsystem operates under the alert dataflow model, sending an email when there is a problem. The two subsystems are linked because the information system in the polling subsystem is the target for the alert subsystem.

3 MonAMI

If all interested parties involved with providing a Grid service were to use the same monitoring infrastructure then a suite of sensors could be developed that would satisfy the monitoring demands. In practise different people are using different monitoring infrastructures. This leads to the awkward situation where there is no one clear monitoring system for which grid middleware developers should provide sensors, no clear monitoring system that people developing monitoring front-ends should look to.

One could view the job of a sensor as having two parts: to gather information about a target and to report this information to some information system. If these two parts were separated, then the collected information could be sent to any number of information systems, based on the sensor's configuration. This would allow the sensor to provide information to any number of interested parties, from local site-administrators through to VO- or Grid-specific monitoring.

We can define a *universal sensor* as being a sensor that can send gathered information to many different (ideally, to all necessary) information systems. To be useful, the sensor should be extensible, so support for additional information systems can be added.

Further, if the sensor configuration is separated into different parts (based on the different interested parties) then providing monitoring information for different people via different information systems becomes tractable. Each group interested in monitoring some grid services can specify their interests independent of others and the universal sensor will capture sufficient information and deliver it according to the different delivery requirements.

The `MonAMI` project[7] provides a framework for developing a universal sensor. It uses a plugin infrastructure to support different information systems, support which can be extended to include additional information systems as needed.

It is important to state that `MonAMI` does not aim to be a complete monitoring solution. Data within any information system has value only if it is then further analysed, for example presented as a graph on a webpage, triggering an email, or used for trend-analysis to predict when computing resources need updating.

3.1 The MonAMI daemon

The `MonAMI` framework currently provides a light-weight monitoring daemon. The advantage of running a daemon process is that many targets can be monitored concurrently. The daemon will automatically aggregate information so that requests for the same monitoring information are consolidated and the impact of monitoring the target is minimised.

This daemon periodically checks the status of *monitoring targets* and reports their status to one or more *reporting targets*. This is the polling model of monitoring dataflow. At the time of writing, support for the asynchronous dataflows (alerts and DIG) is being added.

A target (whether monitoring or reporting) is a specific instance of a *plugin*. The plugin is generic concept (a MySQL server, for example), whilst the target is specific (e.g. the MySQL server running on localhost). The advantage of this approach is that the monitoring plugins need know nothing about how the data is to be reported. Data from a monitoring target can be sent to any number of reporting targets. This also allows `MonAMI` to be extended, both in what information is gathered and to what information systems data is to be sent.

The list of available monitoring and reporting plugins is available on the `MonAMI` project webpage[7] and currently include monitoring the local filesystem, running processes, Apache HTTP[17], MySQL[18] and Tomcat[19] servers. Support exists for reporting plugins, including Nagios[15], Ganglia[16] and MonaLisa[11] in addition to simple file-based logging.

Figure 3 shows a simple configuration for `MonAMI` with the configuration components shown as functional blocks. In this example, routine monitoring services are reported to two information systems (Nagios[15] and Ganglia[16]). There is also a more intensive test, which is conducted far less frequently to reduce its impact. The results of this test are reported to Nagios (to alert sysadmin of undesirable results) and recorded in a log file.

3.2 MonAMI and other components

The grid community has several projects with overlapping goals and implementations. The grid monitoring workgroup of the Global Grid Forum (GGF)[8] has produced a specification for grid-level monitoring: GMA[9]. The R-GMA project[10] has produced a realisation of this work.

Separate from R-GMA, the MonaLisa project[11] has a monitoring infrastructure that is widely used, including a number monitoring targets and end-client applications.

Within WLCG, effort is underway in developing site-level and grid-level functionality tests through the SAM[12] project. This information is aimed towards providing information for site-administrators. For end-users, the Dashboard[13] project aims to provide monitoring information that is easy to navigate.

As previously stated, `MonAMI` does not aim to be a complete solution, but rather to provide useful data to other existing projects. It aims to be part of these larger systems by providing information on key grid services. As a universal sensor, it aims to report to whichever combination of monitoring information systems is currently in use.

Support for additional systems can be added by writing an additional reporting plugin for `MonAMI`. The configuration can be updated to include the extra reporting, allowing data to be reported without affecting any existing monitoring.

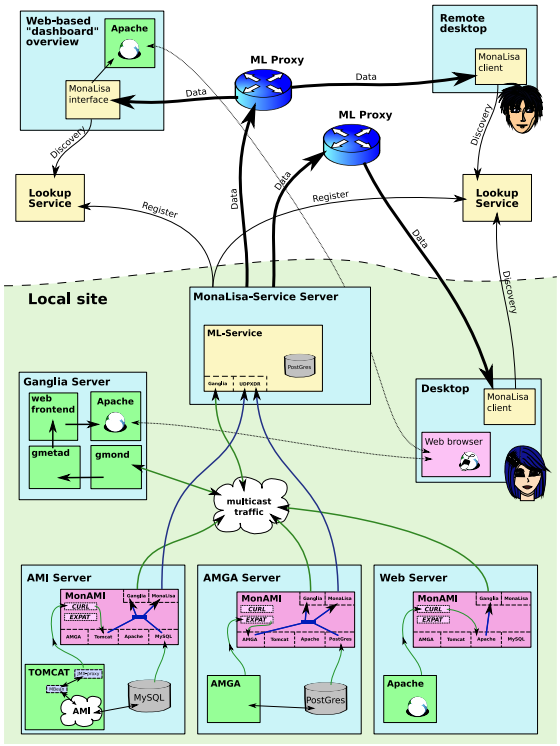


Figure 4: *An example of* `MonAMI` *deployment.*

3.3 Example of MonAMI deployment

`MonAMI` can be deployed in a number of different configurations. It is possible to install a single instance of `MonAMI` and have this daemon monitor services remotely (for services that support remote monitoring). This deployment configuration has minimal deployment impact, but may be inappropriate for services that only allow remote monitoring with insecure password authorisation.

An alternative approach is to install `MonAMI` on all servers that need to be monitored. This approach is more flexible and an example of such a deployment is shown in Figure 4. In this example, monitoring information goes to both the local site administrator (who is using Ganglia[16]) and to a grid-wide information system MonaLisa[11].

Grid-level monitoring is available through the clients provided with MonaLisa, and also through centralised services that query the MonaLisa monitoring information system and provide a webpage summarising the available data (such as Dashboard[13]).

Should additional monitoring be required (such as reporting information to R-GMA or SAM), only an additional file is needed on the monitoring system; this configures `MonAMI` to start reporting the monitoring information to the additional monitoring.

3.4 An explicit use-case

The following use-case illustrates the benefits of using `MonAMI` for monitoring. The use-case assumes monitoring is taking place within WLCG, but applies more generally. The pattern here is kept abstract because currently `MonAMI` is being integrated into the monitoring of different services within various grid efforts and the use-case is generally applicable.

The developers of some grid software need to provide a mechanism that allows site-level monitoring of their service. This is to alert the site-administrators when the local instance of the service has failed or is in danger of failing, and provide some performance metrics to allow those components causing performance bottle-necks to be uncovered.

For providing alerts, supporting Nagios by providing suitable scripts is a choice. Nagios is commonly deployed amongst the larger sites, but its use is not uniform. Providing only Nagios monitoring scripts would support only a subset of sites within the grid. Other site-level monitoring infrastructures exist; supporting all options would require prohibitively large development and support time.

By decoupling the acquisition of data from the delivery of that data, MonAMI allows a single plugin to provide information to any supporting information system (i.e. provide data to any monitoring system MonAMI supports). Development effort can be limited to the single MonAMI plugin and documentation of this plugin's use.

If the monitoring plugin is included within the MonAMI distribution, the service monitoring is expressed as suitable MonAMI configuration files, delivered as an RPM with a dependency on an MonAMI RPM.

If the monitoring plugin is not yet included within a MonAMI release, the service monitoring RPM can include the plugin. MonAMI will include this extra plugin at start-up and provide additional monitoring capability.

The site-administrator connects the monitoring to her existing monitoring infrastructure (provided it is supported by MonAMI). This might require some site-specific adjustments, based on expecting behaviour and site-specific configuration. Nominal, expected values can be published, but the site-administrator might need to adjust these, based on observed usage patterns.

3.5 Anatomy of a plugin

A `MonAMI` plugin is simply a shared object (or a shared library). Under the various GNU/Linux

distributions, it is now common to include support for (and use by default) the ELF file format for executable content. ELF includes support for dynamically loading data. The core part of MonAMI (MonAMI-core) uses the OS provided linker support for loading shared objects to load plugins.

Its important to note that MonAMI-core has no internal list of what plugins are available. Instead, it scans for available plugins at run time and loads those that are available. This allows for additional plugins to be included independently of upgrading the MonAMI distribution.

Each plugin contains a brief overview of the plugin; containing the plugin name, a brief description and what operations the plugin supports.

A plugin can be mentioned within the configuration multiple times, once for each service the plugin is to monitor, or information system to which the plugin is to report. Each mention within the configuration file creates a new target based on the named plugin (as described in 3.1).

The synchronous API requires a plugin to provide two functions (monami_open and monami_close) and at least one of a further two functions (monami_read and monami_write) depending on whether the plugin is providing support for monitoring or reporting activity respectively.

monami_open is called once for each target and allows the allocation of the resources for that target and register with remote services, if necessary. It also allows the plugin to check that the necessary parameters have been provided.

monami_close is called before MonAMI terminates. It provides a clean method of deregistering any allocated resources and removing registration with remote sites, if necessary.

monami_read is called when requesting information from a monitoring plugin. The plugin acquires the current status of the service it is monitoring and provides this information to MonAMI-core.

monami_write is called when data has been collected for the reporting plugin to store or pass on to some information system.

At time of writing, work is underway towards adding support for asynchronous monitoring. This will involve extending the above API to support the two asynchronous dataflow models (Alerts and DIG). The changes aim to be backward compatible, allowing the use of existing plugins without change.

The MonAMI source provides a framework for writing plugins; a basic plugin need do little more that provide three of the above four functions and use a provide macro to create the plugin description.

Further details on how to write plugins is available within the Developer's guide, which documents MonAMI and describes the pedagogical code included in the source distribution.

4 Conclusions

MonAMI aims to provide a framework for developing a universal sensor: a sensor that can report to many different information systems. It allows the monitoring of multiple targets, with information about each target being sent to any number of information systems. MonAMI uses a plugin system so additional monitoring can be supported by adding extra plugins.

For each server, the list of what is to be monitored is separable. This allows the addition of extra monitoring requirements without affecting the existing monitoring.

MonAMI does not aim to be a complete monitoring solution, but rather a building block: a low overhead, extensible method of providing multiple groups of people with the monitoring information they need.

References

[1] The WLCG project (until recently known as LCG).
http://lcg.web.cern.ch/LCG/

[2] Information about how to join WLCG, this includes testing that the WLCG software stack has been installed properly.
http://lcg.web.cern.ch/LCG/Sites/sites.html

[3] *GridPP: Development of the UK Computing Grid for Particle Physics* The GridPP collaboration, 2006 J Phys G: Nuclear and Particle Physics **32** N1-N20 (Technical supplement)

[4] The WLCG "Memorandum of Understanding" document.
http://lcg.web.cern.ch/lcg/C-RRB/MoU/WLCGMoU.pdf

[5] Monitoring within the WLCG is organised through the Grid Operations Centre (GoC).
http://goc.grid-support.ac.uk/gridsite/monitoring/

[6] Within the GridPP project, centralised monitoring is available from the man website.
http://www.gridpp.ac.uk/

[7] The MonAMI project. Aims to be a universal sensor framework.
http://monami.sourceforge.net/

[8] The Global Grid Forum.
http://www.gridforum.org/

[9] Grid Monitoring Architecture specification. Provides information on how compliant grid-wide monitoring should interact.
http://www-didc.lbl.gov/GGF-PERF/GMA-WG/

[10] The R-GMA project: an implementation of GMA specification used within the LCG and EGEE projects.
http://www.r-gma.org/

[11] The MonaLisa project. A widely used grid-wide monitoring system.
http://monalisa.cacr.caltech.edu/monalisa.htm

[12] WLCG Service Availability Monitoring.
https://lcg-sam.cern.ch:8443/sam/sam.py

[13] The Dashboard project; provides a web-based overview of available information.
https://uimon.cern.ch/twiki/bin/view/LCG/ARDA-CMS-Dashboard

[14] The LEMON project, extensive testing for large sites.
http://lemon.web.cern.ch/lemon/index.htm

[15] The Nagios project. An advanced generic tool for triggering alerts based on service availability and performance.
http://www.nagios.org/

[16] The Ganglia project. A hierarchical generic tool for providing graphical monitoring of service performance.
http://ganglia.sourceforge.net/

[17] The Apache HTTP server.
http://httpd.apache.org/

[18] MySQL: perhaps the most popular database.
http://www.mysql.com/

[19] Tomcat: the Apache foundation's Java application server.
http://tomcat.apache.org/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Co-Allocation, Fault Tolerance and Grid Computing

Jon MacLaren,[1] **Mark Mc Keown**,[2] and Stephen Pickles[2]

[1] *Center for Computation and Technology, Louisiana State University, Baton Rouge, Louisiana 70803, United States.*

[2] *Manchester Computing, The University of Manchester, Oxford Road, Manchester M13 9PL.*

Experience gained from the TeraGyroid and SPICE projects has shown that co-allocation and fault tolerance are important requirements for Grid computing. Co-allocation is necessary for distributed applications that require multiple resources that must be reserved before use. Fault tolerance is important because a computational Grid will always have faulty components and some of those faults may be Byzantine. We present HARC, Highly-Available Robust Co-allocator, an approach to building fault tolerant co-allocation services. HARC is designed according to the REST architectural style and uses the Paxos and Paxos Commit algorithms to provide fault tolerance. HARC/1, an implementation of HARC using HTTP and XML, has been demonstrated at SuperComputing 2005 and iGrid 2005.

I. INTRODUCTION

In this paper we discuss the importance of co-allocation to Grid computing. We provide some background on the difficulties associated with co-allocation before presenting HARC, Highly-Available Robust Co-allocator, a fault tolerant approach to co-allocation. The paper also includes a discussion on the problem of fault tolerance and Grid computing. We make the case that a computational Grid will always have faulty components and that some of those faults will be Byzantine [26, 27]. However, we also demonstrate that with suitable approaches it is still possible to make a computational Grid a fault tolerant system.

There are many definitions of Grid computing but recurring themes are: large scale or internet scale distributed computing and sharing resources across multiple administrative domains. The goal of sharing resources between organizations dates back to the ARPANET [33] project and progress towards that goal can be seen in the development of the Internet, the World Wide Web [22] and now computational Grids. While the goals of the ARPANET project are still relevant today the underlying infrastructure has changed, powerful computers have become cheap and plentiful while high performance networks have become pervasive, presenting developers with a different set of challenges and opportunities. We believe that co-allocation is a new challenge, while the falling cost of components makes fault tolerance a new opportunity.

Parallel to the evolution of ARPANET through to Grid computing has been the development of the theory of distributed systems providing us with a deeper understanding of distributed systems and a set of algorithms for building fault tolerant systems. Representational State Transfer [12], REST, is an architectural style for building large scale distributed systems that was used to develop the protocols that make up the World Wide Web. Paxos [28] is a fault tolerant consensus algorithm that can be used to build highly available systems [31]. Paxos Commit [18] is Paxos applied to the distributed transaction commit problem.

HARC uses REST and Paxos to provide a system that is fault tolerant and suitable for a large scale distributed system such as a computational Grid. HARC's focus on fault tolerance is unique among approaches to designing co-allocation services [3, 9, 24, 34, 39].

II. CO-ALLOCATION

Running distributed applications on a computational Grid often requires that the resources needed by the application are available at the same time. The resources may need to be booked (*eg* Access Grid nodes) or they may use a batch submission system (*eg* HPC systems). We define co-allocation as the provision of a set of resources at the same time or at some co-ordinated set of times. Co-allocation can be achieved by making a set of reservations for the required resources with the respective resource providers.

Experience from the award winning TeraGyroid [7] and SPICE [23] projects has shown that support for co-allocation on current production Grids [37, 38] is *ad-hoc* and often requires time consuming direct negotiation with the resource administrators. The resources required by TeraGyroid and SPICE included HPC systems, special high performance networks, high end visualization systems, Access Grid nodes, haptic devices and people. The lack of convenient co-allocation services prevents the routine use of the techniques pioneered by TeraGyroid and SPICE. Without co-allocation computational Grids are limited in the type of applications they can support, and so are limited in their potential. Co-allocation's im-

portance to Grid computing means that it must be a reliable service.

III. FAULT TOLERANCE AND GRID COMPUTING

Whatever definition of Grid computing is used we are lead to two inescapable conclusions: a computational Grid will always have faulty components and some of those faults will be Byzantine.

Computational Grids are internet scale distributed systems, implying large numbers of components and wide area networks. At this scale there will always be faulty components.

The case that a computational Grid will always have faulty components is illustrated by the fact that on average over ten percent of the daily operational monitoring tests, GITS [4], run on the UK National Grid Service, UK NGS [38], report failures. Since the start of the UK NGS a number of the core nodes have been unavailable for days at a time due to planned and unplanned maintenance.

Crossing administrative boundaries raises issues of trust: Can a user trust that a resource provider has configured and administrates the resource properly? Can a resource provider trust that a user will use the resource correctly? Distributed transactions are rarely used between organizations because of a lack of trust; one organization will not allow another organization to hold a lock on its internal databases. Without trust users, resource providers and middleware developers must be prepared for Byzantine fault behaviour: when a component faults but continues to operate in an unpredictable and potentially malicious way. Although Grid computing supports the creation of limited trust relationships between organizations through the concept of the *Virtual Organization* [14] our experience has been that Grids exhibit Byzantine fault behaviour.

To illustrate the point we provide three examples of Byzantine behaviour which we have encountered. The first case involved the UK eScience Grid's MDS2 [2, 10] hierarchy. A GRIS [10] at a site was firewalled preventing GIIS [10] higher up in the MDS2 hierarchy from querying it. The GRIS continued to report that it was publishing information but whenever a GIIS attempted to query it the firewall would block the connection. The GIIS would block waiting for the GRIS to respond causing the whole MDS2 hierarchy to block. The firewall was raised by the site's network administrators. The local firewall on the GRIS server and the GRIS service itself were configured correctly.

The second case involved a job submission node on the TeraGrid [37]. The node consisted of two servers and utilized a DNS round robin to allocate requests to a node. Unfortunately the DNS entries were mis-configured and reported the wrong hostname for one of the nodes causing job submissions to fail randomly. Local testing at the site did not reveal the problem.

The third case involved a resource broker that assigned jobs to computational resources. Occasionally a computational resource would fail in a Byzantine way: it would accept jobs from the resource broker, fail to execute the job but report to the resource broker that the job had completed successfully. The resource broker would continue assigning jobs to the faulty resource until it was drained of jobs.

The examples illustrate the importance of end-to-end arguments in system design — error recovery at the application level is absolutely necessary for a reliable system, and any other error detection or recovery is not logically necessary but is strictly for performance [35]. In each case making the individual components more reliable would not have prevented the problem. Retrofitting reliability to an existing design is very difficult [30].

For co-allocation a very real example of Byzantine fault behaviour occurs when a resource provider accepts a reservation for a resource but at the scheduled time the user, and possibly the resource provider, discover the resource is unavailable.

IV. REST

REST is an architectural style for building large scale distributed systems. It consists of a set of principles and design constraints which were used in designing the protocols that make up the World Wide Web.

We provide a brief description of REST and refer the reader to the thesis [12] for a complete description. REST is based on a client-server model which supports caching and where interactions between client and server are stateless, all interaction state is stored on the client for server scalability. The concept of a resource is central to REST, resources have identity and anything that can have an identity can be a resource. Resources are manipulated through their representations and are networked together through linking — hypermedia is the engine of application state. Together the last set of constraints combine to make up the principle of uniform interface. REST also has an optional constraint for the support of mobile code.

HTTP [11] is an example of a protocol that has been designed according to REST. On the World Wide Web a resource is identified by a URI [6] and clients can retrieve a representation of the resource using a HTTP GET. HTTP can also supply caching information along with the representation to allow

intermediaries to cache the representation. The representation may contain links to other resources creating a network of resources. Clients can change the representation of a resource by replacing the existing representation with a new one, for example using a HTTP PUT. All resources on the World Wide Web have a uniform interface allowing generic pieces of software such as web browsers to interact with them. Web servers are also able to send code, for example JavaScript, to the client to be executed in the browser.

V. PROBLEMS OF CO-ALLOCATION

To illustrate the problems associated with co-allocation we present two possible approaches and discuss their shortcomings. Most existing solutions [3, 9, 24, 34, 39] for co-allocation are based on variations of these approaches.

A. One Phase Approach

In this approach a successful reservation is made in a single step.

The user sends a booking request to each of the Resource Managers (RM) requesting a reservation. The RMs either accept or reject the booking. If one or more of the RMs rejects the booking the user must cancel any other bookings that have been made. This approach has the advantage of being simple but a potential drawback is that the user may be charged each time he cancels a reservation.

Supporting reservations prevents a RM from running the optimal workload on a resource as it has to schedule around the reservations. Even if a reservation is cancelled before the scheduled time it may already have delayed execution of some jobs. RMs may charge for the use of the resource and may charge more for jobs that are submitted via reservation to compensate for the loss in throughput. They may also charge for reservations that are cancelled.

Any charging policy is the prerogative of the RM. Not all RMs may charge and it is unreasonable to make any assumptions about or try to mandate a charging policy. A co-allocation protocol should accommodate the issues associated with charging but cannot depend on RMs supporting charging.

Another potential problem with this approach arises if one of the RMs rejects a booking but the user does not cancel the other reservations that have been made. For example the user may fail before he has had a chance to cancel the reservations. Even if the user recovers he may not have stored sufficient state before failing to cancel the reservations. This is related to the question of trust — the RMs must trust the user to cancel any unwanted reservations.

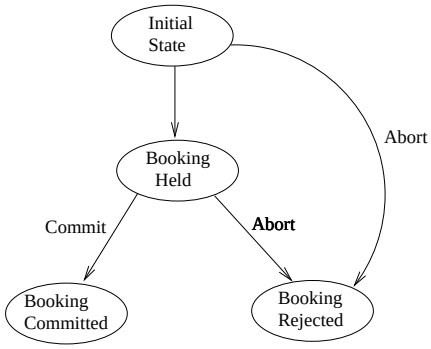


FIG. 1: The state–transition diagram for a RM in the two phase approach. The RM receives the booking request message in the initial state. It can either reject the request and move to the booking rejected state or accept the request and move to the *Booking Held* state. The RM waits in the *Booking Held* state until it receives the *Commit* or *Abort* message from the TM.

A partial solution is to add an intermediary service that is trusted by the user and RM to correctly make and cancel reservations as necessary. The user sends the set of bookings he requires to the intermediary which handles all the interactions with the RMs, cancelling all the reservations if the purposed schedule is unacceptable to any of the RMs. The intermediary is still a single point of failure unless it can be replicated.

B. Two Phase Approach

In this approach a successful reservation is made in two steps.

To resolve the problem of RMs charging for cancelling a reservation we introduce a new state for the RM, *Booking Held*. The client can cancel a reservation in the *Booking Held* state without being charged. The approach is similar to the two phase commit [17] algorithm for distributed transactions but does not necessarily have the ACID (Atomicity, Consistency, Isolation and Durability [19]) properties associated with two phase commit.

A process, the Transaction Manager (TM), tries to get a group of RMs to accept or reject a set of reservations. In the first phase the TM sends the booking requests to the RMs, the RMs reply with a *Booking Held* message if the request is acceptable or an *Abort* message if the request is unacceptable. If all the RMs reply with *Booking Held* the TM sends a *Commit* message to the RMs and the reservations are committed. Otherwise the TM sends an *Abort* message to the RMs and no reservations are made.

Fig. 1 shows the state transitions for the RM in the two phase approach.

Unfortunately the two phase approach can block. If the TM fails after the first phase but before starting the second phase, before sending the *Commit* or *Abort* message, the RMs are left in the *Booking Held* state until the TM recovers. While in the *Booking Held* state the RMs may be unable to accept reservations for the scheduled time slot and the resource scheduler may be unable to run the optimal work load on the resource.

The blocking nature of the two phase approach may be acceptable for certain scenarios depending on how long the TM is unavailable and providing it recovers correctly. To overcome the blocking nature of the two phase approach requires a three phase protocol [36] such as Paxos.

Another potential problem with the two phase approach is that RMs must support the *Booking Held* state. A solution is for RMs that do not support the *Booking Held* state to move straight to the *Booking Committed* state if the booking request is acceptable in the first phase. For the second phase they can ignore a *Commit* message and treat an *Abort* message as a cancellation of the reservation for which the user may be charged. This a relaxation of the consistency property of two phase commit.

It is also possible to relax the isolation property of two phase commit. If a RM receives a booking request while in the *Booking Held* state it can reject the request, advising the client that it is holding an uncommitted booking which may be released in the future. This prevents deadlock [19], when two users request the same resources at the same time.

The role of the TM can be played by the user but should be carried out by a trusted intermediary service.

VI. CO-ALLOCATION AND CONSENSUS

Co-allocation is a consensus problem. Consensus is concerned with how to get a set of processes to agree a value [26]. Distributed transaction commit, of which two phase commit is one approach, is a special case of consensus were all the processes must agree on either commit or abort for a transaction. In the case of co-allocation the RMs must agree to either accept or reject a purposed schedule.

Consensus is a well understood problem with over twenty five years of research [13, 26–28]. For example it has been shown that distributed consensus is impossible in an asynchronous system with a perfect network and just one faulty processor [13]. In an asynchronous system it is impossible to differentiate between a processor that is very slow and one that has failed. To handle faults requires either fault detectors or partial synchrony. Consensus is an important problem because it can be used to build replicas: start a set of deterministic state machines in the same state and use consensus to make them agree on the order of messages to process [25, 31].

Paxos is a well known fault tolerant consensus algorithm. We present only a description of its properties and refer the reader to the literature [28, 29, 31] for a full description of the algorithm. In Paxos a leader process tries to guide a set of acceptor processes to agree a value. Paxos will reach consensus even if messages are lost, delayed or duplicated. It can tolerate multiple simultaneous leaders and any of the processes, leader or acceptor, can fail and recover multiple times. Consensus is reached if there is a single leader for a long enough time during which the leader can talk to a majority of the acceptor processes twice. It may not terminate if there are always too many leaders. There is also a Byzantine version of Paxos [8] to handle the case were acceptors may have Byzantine faults.

Paxos Commit [18] is the Paxos algorithm applied to the distributed transaction commit problem. Effectively the transaction manager of two phase commit is replaced by a group of acceptors — if a majority of acceptors are available for long enough the transaction will complete. Gray and Lamport [18] showed that Paxos Commit is efficient and has the same message delay as two phase commit for the fault free case. They also showed that two phase commit is Paxos Commit with only one acceptor.

Though it may be possible to apply Paxos directly to the co-allocation problem by having the RMs act as acceptors we chose to use Paxos Commit instead. The advantage Paxos Commit has over using Paxos directly for co-allocation is that the role played by the RMs is no more complex than in the two phase approach. Limiting the role of the RM makes it more acceptable to resource providers and reduces the possibilities for faulty behaviour from the RM.

VII. HARC

The HARC approach to co-allocation is similar to the two phase approach described in Section II except the TM is replaced with a set of Paxos acceptors. The user sends a booking request to an acceptor who first replicates the message to the other acceptors using Paxos and then it uses Paxos Commit to make the reservations with the RMs. HARC terminates once it has decided to commit or abort a set of reservations. Users and RMs may modify or cancel reservations after HARC has terminated, but this is outside the scope of HARC.

The aims of HARC are deliberately limited. It does not address the issues of how to choose the op-

timal schedule; how to manage a set of reservations once they have been made; how to negotiate quality of service with the resource provider or how to support compensation mechanisms when a resource fails to fulfil a reservation or when a user cancels a reservation. HARC is designed so that other services and protocols can be combined with it to solve these problems.

A. Choosing a Schedule

The RM advertises the schedule when a resource may be available through a URI. The user retrieves the schedule using a HTTP GET. The information retrieved is for guidance only, the RM is under no obligation by advertising it. It is the RM's prerogative as to how much information it makes available, it may choose not to advertise a schedule at all. The RM may supply caching information along with the schedule using the cache support facilities of HTTP. The caching information can indicate when the schedule was last modified or how long the schedule is good for. The RM may also support conditional GET [11] so that clients do not have to retrieve and parse the schedule if it hasn't changed since the last retrieval. From the schedules for all the resources the user chooses a suitable time when the resources he requires might be free.

A co-scheduling service for HARC could use HTTP conditional GET to maintain a cache of resource schedules from which to create co-schedules for the user. The co-scheduling service would have to store only soft state making it easy to replicate.

B. Submitting the Booking Request

The user constructs a booking request which contains a sub-request for each resource that he wants to reserve. The booking request also contains an identifier chosen by the user, the UID. The combination of the identity of the user and the UID should be globally unique. The user sends the booking request to an acceptor using a HTTP POST. This acceptor will act as the leader for the whole booking process unless it fails in which case another acceptor will take over.

The leader picks a transaction identifier, the TID, for the booking request from a set of TIDs that it has been initially assigned. Each acceptor has a different range of TIDs to choose from to prevent two acceptors trying to use the same TID. The acceptor effectively replicates the message to the other acceptors by having Paxos agree the TID for the message. This instance of Paxos agrees a TID for the combination of the user's identity and the user chosen UID. If the user resends the message to the acceptor or to any other acceptor he will receive the same TID.

The user should continue submitting the request to any of the acceptors until he receives a TID — the submission of the booking request is idempotent across all the acceptors.

If a user reuses a UID he will get the TID of the previous booking request associated with that UID. To avoid reusing a UID the user can record all the UIDs he has previously used, however a more practical approach is to randomly choose a UID from a large namespace. Two users can use the same UID as it is the combination of the user's identity and the UID that is relevant.

The TID is a RequestURI [11]. The user can use a HTTP GET with the TID to retrieve the outcome of the booking request from any of the acceptors. The TID is returned to user using the HTTP Location header in the response to the POST containing the booking request. The HTTP response code is 201 indicating that a new resource has been created. The acceptor should return a 303 HTTP response code if a TID has already been chosen for the request to allow the user to detect the case when a UID may have been reused.

C. Making the Reservations

After the TID has been chosen the leader uses Paxos Commit to make the bookings with the RMs.

The booking request is broken down into the sub-requests and the sub-requests are sent to the appropriate RM using HTTP POST. The TID is also sent to the RM as the Referer HTTP header in the POST message.

The RM responds with a URI that will represent the booking local to the RM and whether the booking is being held or has been rejected. It also broadcasts this message to all the other acceptors.

As in the Paxos Commit algorithm the acceptors decide whether the schedule chosen by the user is acceptable to all the RMs or has been rejected by any of the RMs. The leader informs each RM whether to commit or abort the booking using a HTTP PUT on the URI provided by the RM.

It is the RM's obligation to discover the outcome of the booking. If the RM does not receive the commit or abort message it should use a HTTP GET with the TID on any of the acceptors to discover the outcome of the booking. Once the acceptors have made a decision it can be cached by intermediaries.

A HTTP GET on the TID returns the outcome of the booking request and a set of links to the reservations local to each RM. Detailed information on a reservation at a particular RM can be retrieved by

using a HTTP GET on the link associated with that reservation.

The user can cancel the reservation through the URI provided by the RM and the RM can advertise that it has cancelled the reservation using the same URI. The user should poll the URI to monitor the status of the reservation, HTTP HEAD or conditional GET can be used to optimize the polling.

There is a question of what happens if a RM decides to unilaterally abort from the *Booking Held* state which is not allowed in two phase commit or Paxos Commit. Since HARC terminates when it decides whether a schedule should be committed or aborted we can say that the RM cancelled the reservation after HARC terminated. This is possible because there is only a partial ordering of events in a distributed system [25]. The user can only find out that the RM cancelled the reservation after HARC terminated. The onus is on the user to monitor the reservations once they have been made and to deal with any eventualities that may arise.

D. HARC Actions

HARC supports a set of actions for making and manipulating reservations: Make, Modify, Move and Cancel. Make is used to create a new reservation, Modify to change an existing reservation (*eg* to change the number of CPUs requested), Move to change the time of a reservation and Cancel to cancel a reservation. A booking request can contain a mixture of actions and HARC will decide whether to commit or abort all of the actions. The HARC actions provide the user with flexibility for dealing with the case of a RM cancelling a reservation, for example he could make a new reservation on another resource at a different time and move the existing reservations to the new time.

A third party service could monitor reservations on the behalf of the user and deal with any eventualities that arise using the HARC actions in accordance to some policy provided by the user.

E. Security

The complete booking request sent to the acceptor is digitally signed [5] by the user with a X.509 [1] certificate. The acceptors use the signature to discover the identity of the user. Individual sub-requests are also digitally signed by the user so that the RMs can verify that the sub-request came from an authorized user. The acceptors also digitally sign the sub-requests before passing them to the RMs so that the RM can verify that the sub-request came from a trusted acceptor. If required a sub-request can be digitally encrypted [21] by the user so that the acceptors cannot read it.

A HTTP GET using the TID returns only the outcome of a booking request and a set of links to the individual bookings so there is no requirement for acceptors to provide access control to this information. The individual RMs control access to any detailed information on reservations they hold.

VIII. HARC AVAILABILITY

HARC has the same fault tolerance properties as Paxos Commit which means it will progress if a majority of acceptors are available. If a majority of acceptors are not available HARC will block until a majority is restored, since blocking is unacceptable we define this as HARC failing. To calculate the MTTF for HARC we use the notation and definitions from [19].

The probability that an acceptor fails is $1/MTTF$ were MTTF is the Mean Time To Failure of the acceptor. The probability that an acceptor is in a failed state is approximately $MTTR/MTTF$, were MTTR is the Mean Time To Repair of the acceptor.

Given $2F+1$ acceptors, the probability that HARC blocks is the probability that F acceptors are in the failed state and another acceptors fails.

The probability that F out of the $2F+1$ acceptors are in the failed state is:

$$\frac{(2F+1)!}{F!(F+1)!}\left(\frac{MTTR}{MTTF}\right)^F \tag{1}$$

The probability that one of the remaining $F+1$ acceptors fails is:

$$(F+1)\left(\frac{1}{MTTF}\right) \tag{2}$$

The probability that HARC will block is the product of (1) and (2):

$$\frac{(2F+1)!}{(F!)^2}\left(\frac{MTTR^F}{MTTF^{F+1}}\right) \tag{3}$$

The MTTF for HARC is the reciprocal of (3):

$$MTTF_{HARC} \approx \frac{(F!)^2}{(2F+1)!}\left(\frac{MTTF^{F+1}}{MTTR^F}\right) \tag{4}$$

Using 5 acceptors, $F=2$, each with a MTTF of 120 days and a MTTR of 1 day, $MTTF_{HARC}$ is approximately 57,600 days, or 160 years.

IX. BYZANTINE FAULTS AND HARC

HARC is based on Paxos Commit which assumes the acceptors do not have Byzantine faults but which

can happen in a Grid environment. We believe that acceptors can be implemented as fail–stop [19] services that are provided as part of a *Virtual Organization's* role of creating trust between organizations. Provision of trusted services is one way of realizing the *Virtual Organization* concept. If Byzantine fault tolerance is necessary then Byzantine Paxos [8] could be used in HARC.

If a HARC acceptor does have a Byzantine fault it cannot make a reservation without a signed request from a user. HARC has been designed to deal with Byzantine fault behaviour from users and RMs.

No co-allocation protocol can guarantee that the reserved resources will actually be available at the scheduled time. HARC is a fault tolerant protocol for making a set of reservations, it does not attempt to make any guarantees once the reservations have been made. The user may be able to deal with the situation were a resource is unavailable at the scheduled time by booking extra resources that can act as backup. This is an application specific solution and again illustrates the importance of end-to-end arguments [35]. HARC provides a fault tolerant service to the user but fault tolerance is still necessary at the application level.

X. HARC/1

HARC/1, an implementation of HARC, has been successfully demonstrated at SuperComputing 2005 and iGrid 2005 [20] where it was used to co-schedule ten compute and two network resources. HARC/1 is implemented using Java with sample RMs implemented in Perl. HARC/1 is available at `http://www.cct.lsu.edu/personal/maclaren/CoSched`.

XI. CONCLUSION

As the size of a distributed system increases so to does the probablity that some component in the system is faulty. However, if a fault tolerant approach is applied then increasing the size of the system can mean that the reliability of the overall system improves. Just as Beowulf systems built out of commodity components have displaced expensive supercomputers so clusters of PCs are displacing expensive mainframe type systems [15]. HARC demonstrates how important services can be made fault tolerant to create a fault tolerant Grid.

HARC has been demonstrated to be a secure, fault tolerant approach to building co-allocation services. It has also been shown that the user and RM roles in HARC are simple. The state–transitions for the RM in HARC are the same as for the two phase approach illustrated in Fig. 1. The only extra requirement for the RM over the two phase approach is that it must broadcast a copy of its *Booking Held* or *Abort* message to all acceptors.

The use of HTTP contributes to the simplicity of HARC. HTTP is a well understood application protocol with strong library support in many programming languages. HARC demonstrates through its use of HTTP and URIs the effectiveness of REST as an approach to developing Grid services.

HARC's functionality can be extended by adding other services, for example to support co-scheduling and reservation monitoring. Extending functionality by adding services, rather than modifying existing services, indicates good design and a scalable system in accordance to REST.

The opaqueness of the sub-requests to the acceptors means that HARC has the potential to be used for something other than co-allocation. For example a HARC implementation could be used as an implementation of Paxos Commit to support distributed transactions.

XII. ACKNOWLEDGEMENTS

The authors would like to thank Jim Gray, Savas Parastatidis and Dean Kuo for discussion and encouragement. The work is supported in part through NSF Award #0509465, "EnLIGHTened Computing".

[1] C. Adams and S. Farrell. Internet X.509 Public Key Infrastructure Certificate Management Protocols. IETF RFC 2510, 1999.

[2] R. Allan *et al.* Building the e-Science Grid in the UK: Grid Information Services. Proceedings of UK e-Science All Hands Meeting. 2003.

[3] A. Andrieux *et al.* Web Services Agreement. Draft GGF Recommendation, September 2005.

[4] D. Baker and M. Mc Keown. Building the e-Science Grid in the UK: Providing a software toolkit to enable operational monitoring and Grid integration. Proceedings of UK e-Science All Hands Meeting. 2003.

[5] M. Bartel *et al.* XML-Signature Syntax and Processing. W3C Recommendation, February 2002.

[6] T. Berners-Lee, R. Fielding and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. IETF RFC 3986, 2005.

[7] R. Blake *et al.* The teragyroid experiment—supercomputing 2003. Scientific Computing,

13(1):1-17, 2005.

[8] M. Castro and B. Liskov. Practical Byzantine fault tolerance. Proceedings of 3rd OSDI, New Orleans, 1999.

[9] K. Czajkowski et al. A protocol for negotiating service level agreements and coordinating resource management on distributed systems. Proceedings of 8th International Workshop on Job Scheduling Strategies for Parallel Processing, eds D Feitelson, L. Rudolph and U. Schwiegelshohn, Lecture Notes in Computer Science, 2537, Springer Verlag, 2002.

[10] K. Czajkowski *et al.* Grid Information Services for Distributed Resource Sharing. Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, 2001.

[11] R. Fielding *et al.* Hypertext Transfer Protocol – HTTP/1.1, IETF RFC 2616, 1999.

[12] R. Fielding. Architectural Styles and the Design of Network-based Software Architectures. PhD thesis. University of California, Irvine, 2000.

[13] M. Fischer, N. Lynch, and M. Paterson. Impossibility of distributed consensus with one faulty process. Journal of the ACM, 32(2), 1985.

[14] I. Foster, C. Kesselman and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 15(3), 2001.

[15] S. Ghemawat, H. Gobioff, and S. Leung. The Google Filesystem. 19th ACM Symposium on Operating Systems Principles, Lake George, NY, October, 2003.

[16] M. Gudgin *et al.* SOAP Version 1.2 Part 1: Messaging Framework. W3C Recommendation, June 2003.

[17] J. Gray. Notes on data base operating systems. In R. Bayer, R. Graham, and G. Seegmuller, editors, Operating Systems: An Advanced Course, volume 60 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, New York, 1978.

[18] J. Gray and L. Lamport. Consensus on Transaction Commit. Microsoft Research Technical Report MSR-TR-2003-96, 2005.

[19] J. Gray and A. Reuter. Transaction Processing: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[20] A. Hutanu *et al.* Distributed and collaborative visualization of large data sets using high-speed networks. Submitted to proceedings of iGrid 2005, Future Generation Computer Systems. The International Journal of Grid Computing: Theory, Methods and Applications, 2005.

[21] T. Imamura, B. Dillaway and E. Simon. XML Encryption Syntax and Processing. W3C Recommendation, December 2002.

[22] I. Jacobs and N. Walsh. Architecture of the World Wide Web, Volume One. W3C Recommendation, December 2004.

[23] S. Jha *et al.* Spice: Simulated pore interactive computing environment – using grid computing to understand dna translocation across protein nanopores embedded in lipid membranes. Proceedings of the UK e-Science All Hands Meetings, 2005.

[24] D. Kuo and M. Mc Keown. Advance reservation and co-allocation for Grid Computing. In First International Conference on e-Science and Grid Computing, volume e-science, IEEE Computer Society Press, 2005.

[25] L. Lamport. Time, Clocks and the Ordering of Events in a Distributed System. Communications of the ACM 21, 7, 1978.

[26] L. Lamport, M. Pease and R. Shostak. Reaching Agreement in the Presence of Faults. Journal of the Association for Computing Machinery 27, 2, 1980.

[27] L. Lamport, M. Pease and R. Shostak. The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems 4, 3, 382-401, 1982.

[28] L. Lamport. The Part-Time Parliament. ACM Transactions on Computer Systems 16, 2, 133-169, 1998.

[29] L. Lamport. Paxos Made Simple. ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001) 18-25, 2001.

[30] B. Lampson. Hints for computer system design. ACM Operating Systems Rev. 17, 5, pp 33-48, 1983.

[31] B. Lampson. How to build a highly available system using consensus. In Distributed Algorithms, ed. Babaoglu and Marzullo, Lecture Notes in Computer Science 1151, Springer, 1996

[32] E. Rescorla. HTTP Over TLS. IETF RFC 2818, 2000.

[33] L. Roberts. Resource Sharing Computer Networks. IEEE International Conference, New York City. 1968.

[34] A. Roy. End-to-End Quality of Service for High-end Applications. PhD thesis. University Of Chicago, Illinois, 2001.

[35] J. Saltzer, D. Reed and D. Clark. End-to-end arguments in system design. Proceedings of the 2nd International Conference Distributed Computing Systems, Paris, 1981.

[36] D. Skeen. Nonblocking commit protocols. In SIGMOD '81: Proceedings of the 1981 ACM SIGMOD International Conference on Management of Data. ACM Press, 1981.

[37] TeraGrid. `http://www.teragrid.org/`.

[38] UK National Grid Service. `http://www.ngs.ac.uk/`.

[39] K. Yoshimoto, P. Kovatch and P. Andrews. Co-scheduling with user–settable reservations. In Job Scheduling Strategies for Parallel Processing, eds E. Frachtenberg, L. Rudolph and U. Schwiegelshohn, Lecture Notes in Computing Science, 3834, Springer, 2005.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Designing a Java-based Grid Scheduler using Commodity Services

Patrick Wendel
patrick@inforsense.com
InforSense
London

Arnold Fung
arnold@inforsense.com
InforSense
London

Moustafa Ghanem
mmg@doc.ic.ac.uk
Computing Department
Imperial College
London

Yike Guo
yg@doc.ic.ac.uk
Computing Department
Imperial College
London

Abstract

Common approaches to implementing Grid schedulers usually rely directly on relatively low-level protocols and services, to benefit from better performances by having full control of file and network usage patterns. Following this approach, the schedulers are bound to particular network protocols, communication patterns and persistence layers. With the availability of standardized high-level application hosting environments providing a level of abstraction between the application and the resources and protocols it uses, we present the design and the implementation necessary to build a Java-based, protocol agnostic, scheduler for Grid applications using commodity services for messaging and persistence. We present how it can be deployed following two different strategies, either as a scheduler for a campus grid, or a scheduler for a wide-area network grid.

1 Motivation

This project was started as part of the development of the Discovery Net platform[1], a workflow-based platform for the analysis of large-scale scientific data.

The platform's architecture consists of one or more workflow execution servers and a workflow submission server tightly coupled with an interactive client tool for building, executing and monitoring. Thus the client tool benefits from the ability to communicate complex objects as well as code with the other components of the system and allow rich interaction between these components. Interoperability of the workflow server with other services within a more loosely-coupled Grid architecture is enabled by providing a set of stateless services accessible using Web Services protocols, although for a subset of the functionalities.

As well, the platform relies on the services of a Java application server for providing a hosting environment for the workflow activities to be executed. This environment can, for instance, provide the activity with support for authentication and authorisation management or logging.

However, such an approach has the drawback of complicating the integration with schedulers based on native process submission, which is usually the case, as in our case each workflow execution has to run within a hosting Java-based environment that provides a different set of services above the operating system. Equally, it is difficult to reuse the clustering features usually provided by Java application servers as they are designed for short executions, usually for transaction-based web applications, over a close cluster of machines.

2 Approach

Instead of trying to integrate schedulers based around command-line tools and native processes, the approach is to build a generic scheduler for long-running tasks using the services provided by the hosting environment. In particular, the availability of a messaging service providing both point-to-point and publish/subscribe models, a container-managed handling of the persistence of long-lived objects as well as the ability to bind object types to specific point-to-point message services, are of interests for building the scheduler.

It follows from that approach that the implementation:

- only requires a few classes,
- does not access I/O and resources directly,
- is network protocol agnostic,
- is agnostic to the Java application server it runs atop.

The scheduling policy resides in the messaging service's handling of the subscribers to its point-to-point model. The service provider used in the experiment allows configuring and extending that handling, thus making it possible to use various scheduling algorithm or services. As an example, the Sun Grid Engine was used to find out what resource the scheduler should choose.

3 Container services

The design is based around a set of four standard mechanisms available in Java-based application server, part of the Enterprise Java Beans[3] (EJB) and Java Messaging System[4] (JMS) specifications, as shown on Figure 1.

3.1 Stateless Objects

Stateless remote objects, also known as *Stateless Session Beans*, have a very simple lifecycle as they cannot keep any state. The container then provides support to access these objects following several protocols:

- RMI/JRMP: For Java-based systems, thus allowing to exchange any serialiseable Java object.
- RMI/IIOP: To support CORBA-IIOP interoperability
- SOAP/WSDL: To support Web Services interoperability

3.2 Persistence Management for Stateful Objects

This service allows the container to the lifecycle and the persistence of stateful objects. The objects are mapped into a relational database using a predefined mapping and the container is responsible for making sure of the consistency between the database instance and the object in memory. This service is provided as *Container-Managed Persistence for Entity Beans*.

3.3 Messaging

JMS is a messaging service that supports both point-to-point model using *Queue* objects and publish/subscribe model using *Topic* objects.

JMS providers are responsible for the network protocol that they use to communicate and deliver messages. In particular, the service we used JBossMQ has support for the following communication protocols:

- RMI/JRMP: Allows faster communication by pushing the notifications to the subscriber, but requires the subscriber to be able to export RMI objects. Being able to export RMI objects adds constraints on the network architecture as it means that the machine that exports the object must know the IP address or name by which it can be reached by the caller. This is the main reason why such protocol cannot be used easily for WAN deployments if the subscribers for the message service is behind a firewall or belongs to a network using NAT.
- HTTP: On the subscriber-side, the messaging service pulls regularly information from the messaging provider. This approach solves the issue discussed above but is not as efficient as sending the notification as it happens.

3.4 Message-driven Objects

Associated with the messaging service, special object types can be registered to be instantiated to handle messages coming to a *Queue* object (point-to-point model), thus removing the need of subscribers to act as factories for the actual instances that will deal with the processing of the object from the queue.

3.5 Security Modules

The authentication and authorisation mechanism for Java containers[8] (JAAS) supports the definition of authentication policies as part of the configuration of the containers and the descriptors of the application instead of being coupled to the application code itself. It also allows defining authorisation information such as the roles associated with a user. Modules can be defined to authenticate access to components using most standard mechanisms such as LDAP-based authentication infrastructures, NT authentication, UNIX authentication, as well as support for Shibboleth[10]. In our application, this facility is used to enable secure propagation of authentication information from the submission server to the execution server.

4 Design

4.1 Architecture

The scheduler was designed to be applied to workflow executions. One of the differences between the submission of workflows and the submission of executables and scripts invoked through command lines, as is often the case for job submission, is the size of the workflow description. Potentially, the workflow can represent a complex process, and its entire description needs to be submitted for execution. The system must therefore ensure that the workflow is reliably stored in a database before performing the execution, as the risks of failures at that stage are greater.

The overall architecture is shown in Figure 2. Both Web and thick clients talk to the *TaskManagement* service, a stateless service implemented by a stateless session bean for job submission, control and some basic level of monitoring. The client also connects to the messaging service to receive monitoring information about the execution.

The *TaskManagement* service is hosted by a container that also provides hosting to the *JobEntity* bean, a container-managed persistence entity bean stored in a

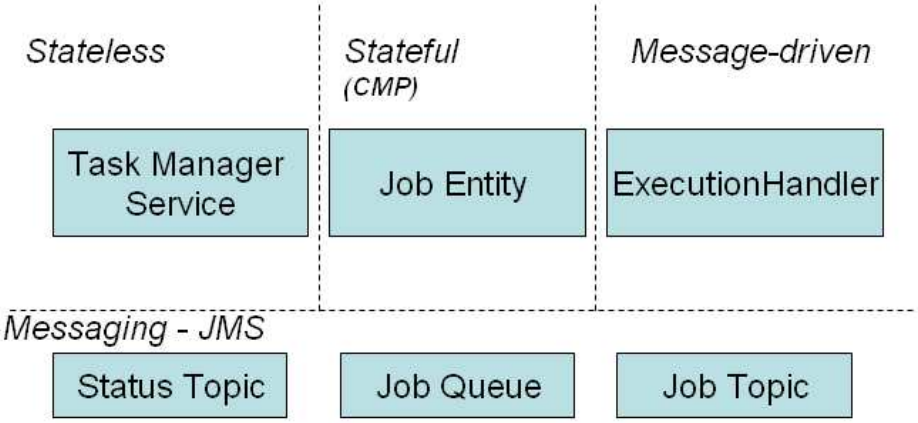


Figure 1: Container Services

common persistent storage, and access to the JMS service providers for the *Job* queue, the *Job* topic and the *Status* topic. The server hosting this service is also called submission server.

A *JobEntity* instance has the main following variables: unique ID, execution status, workflow definition, workflow status information, last status update time, start date, end date, user information.

Execution servers host a pool of message driven *ExecutionHandler* beans which, on receiving a message from the *Job* queue, subscribe to messages added to the *Job* topic. The pool size represents the maximum number of tasks that the execution server allows to be processed concurrently. The container hosting the execution server also has access to the same persistent storage and messaging service providers as the submission server and so can host instances of *Job* entities.

4.2 Submission

The following sequence of events happen when a workflow is submitted for execution (See Figure 3):

1. The client submits the workflow to the *TaskManagement* service
2. The service then creates a new *JobEntity* object, which is transparently persisted by the container in the database
3. It then publishes a request for execution to the*Job* queue and returns the ID of the *JobEntity* to the caller.
4. That execution is picked up by one of the *ExecutionHandlers* of any execution server, following the allocation policy of the JMS provider.
5. The *ExecutionHandler* subscribe to the *Job* topic, selecting only to be notified of messages related to the ID of the *JobEntity* it must handle.
6. The *ExecutionHandler* then instantiates the *JobEntity* object and starts its execution.

4.3 Control

The following sequence of events happen when the user sends a control command to the execution handler, such as *pause*,*resume*,*stop* or *kill* (See Figure 4):

1. The *TaskManagement* service receives the request for a control command to a given JobEntity ID.
2. If the request to execute that *JobEntity* is not in the *Job* queue and its state is *running* then it posts the control request to the *Job* topic.
3. The listening *ExecutionHandler* receives the notification and performs the control action on the *JobEntity* which will accordingly modify its execution status.

4.4 Monitoring

As the scheduler is used to execute workflows which must be monitored from a client tool, the monitoring mechanism needs to support relatively large workflow status information that could include complex activity specific objects describing the status of each running activity in the workflow. The sequence of events for monitoring the execution is as follows (See Figure 5):

1. The *ExecutionHandler* for a running *JobEntity* regularly requests the latest workflow status information from the running workflow.

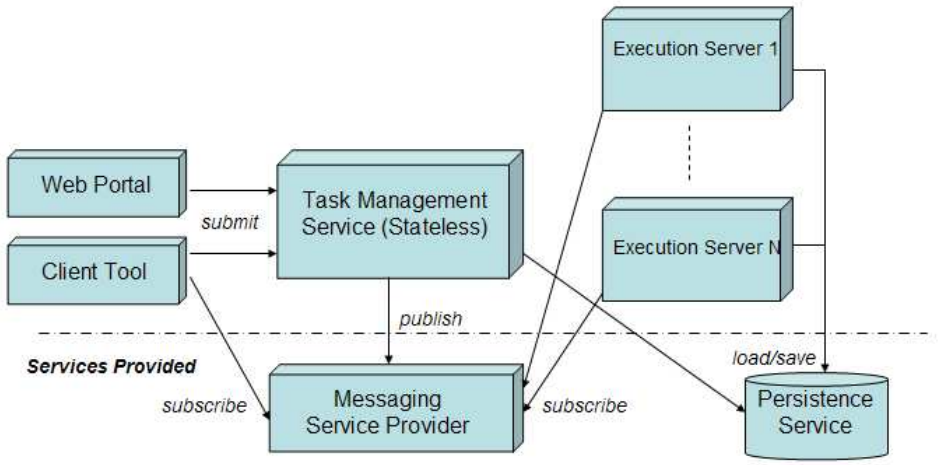


Figure 2: Architecture

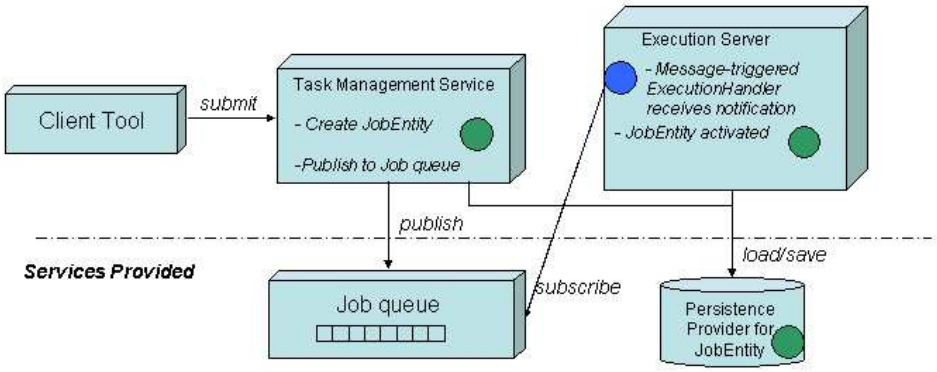


Figure 3: Job Submission

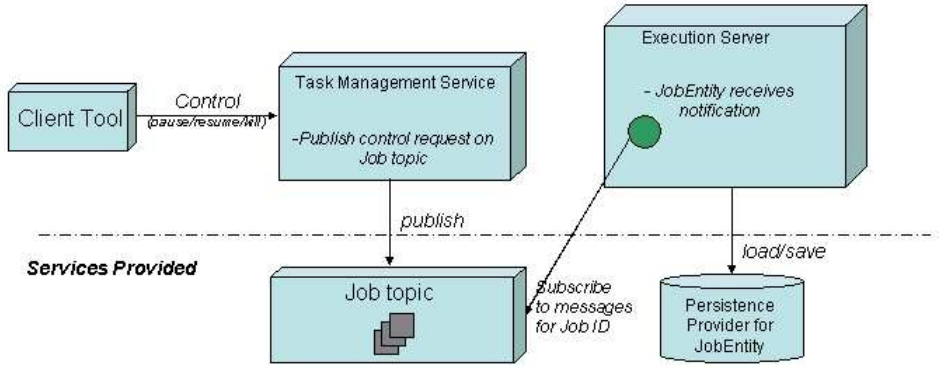


Figure 4: Job Control

2. If that status has changed since the last status update, the state of the *JobEntity* is updated and the new execution status and workflow status information is submitted to the *Status* topic.

3. While the client tool is started, it will be subscribing to publications on the *Status* topic, for the tasks that have been submitted by the current user, and will therefore receive the notification and associated status information.

The policy followed by the *ExecutionHandler* to schedule the request for the latest workflow status information, is based on a base period and a maximum period. The status is requested according to the base period, except if the status has not changed in which case the update period is doubled, up to the maximum period.

4.5 Failure detection

One problem with the de-coupled architecture presented, is that there is no immediate notification that an execution server has failed, as it only communicates through the messaging service which does not provide by default information to the application about its subscribers. The approach used to detect failures of the execution server is to check regularly, on the server hosting the *TaskManagement* service, the last status update time of all the running *JobEntity*. If that update time is significantly above the maximum update period, then that job is stopped, killed if necessary, and restarted.

4.6 Security

In order to make sure that workflows run in the correct context and has the correct associated roles and authorization, the *ExecutionHandler* needs to impersonate the user who submitted the workflow. This is implemented by a specific JAAS module that enables that impersonation to happens for a particular security policy used by the *ExecutionHandler* to login.

4.7 Scheduling Policy

The scheduling policy is defined by the way the JMS service provider decides which *ExecutionHandler* should receive requests added to the Job queue. Several policies are provided by default with the JMS provider that was used, in particular we used a simple round-robin policy at first.

In order to integrate at that stage with a resource management tool such as the Grid Engine[11], the scheduling policy was extended. To find out which subscriber to use, we assumed that the set of execution servers was the same as the set of resources managed by the grid engine and submitted a request to execute the command `hostname` and use the information returned to choose the relevant subscriber.

Although these policies can be sufficient in general, for our application to workflow execution, another policy was created to make sure that we use the resource that holds any intermediate results that have already been processed in the workflow, in order to optimise its execution. In this case, the policy has to check the workflow description associated with the request, to find out any intermediate results associated with any activity, and then decide to use the corresponding server if possible.

5 Deployment

The persistence service provider used is HSQL[5]. It provides support for the persistence of basic types as well as Java objects. The messaging service is JBossMQ[7]. Messages are stored if needed in the same HSQL database instance used for the persistence of container managed entity objects.

5.1 Campus Grid scheduler

The first deployment of the scheduler uses protocols that are only suitable over an open network without communication restrictions or network address translation (NAT) in some parts. This is usually the case for deployment inside an organisation. The RMI protocol can be used for simplicity and efficiency, as well as direct connection and notification of the client tool can be performed without the risk of network configuration issues. This setup is described in Figure 6

5.2 Scheduling over WAN

The second deployment of the scheduler uses HTTP tunnelling for method calls and HTTP-based polling mechanism for queue and topic subscribers as shown on Figure 7. The main advantages are that the client does not have to have an IP address accessible by the messaging service, as there is no direct call-back. This means that although the client tool, in our application, performs rich interactions with the workflow, it does not have to be on the same network as the task management server.

The execution servers also do not need to have a public IP, which makes it theoretically possible, given the right software delivery mechanism for the execution server, to use the scheduler in configuration such as supported by the SETI@Home[2] scheduler.

Other configurations need to be modified to support such deployment. In particular the values for *time out* and *retries* values for connections to the persistence manager and the messaging service need to be increased, as network delays or even network failures are more likely.

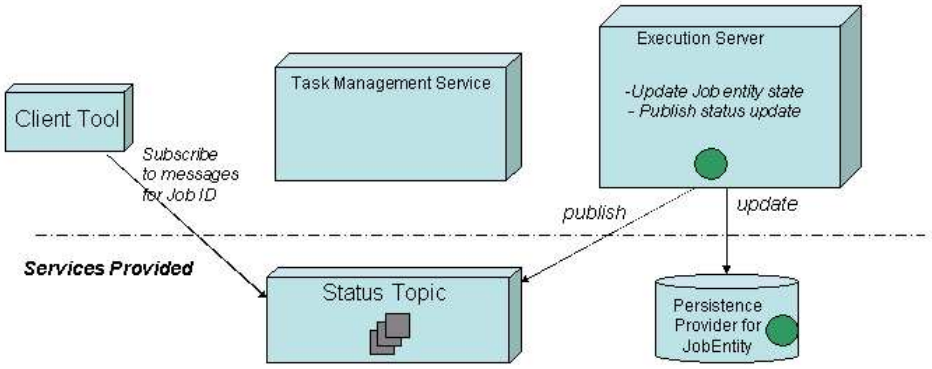


Figure 5: Job Monitoring

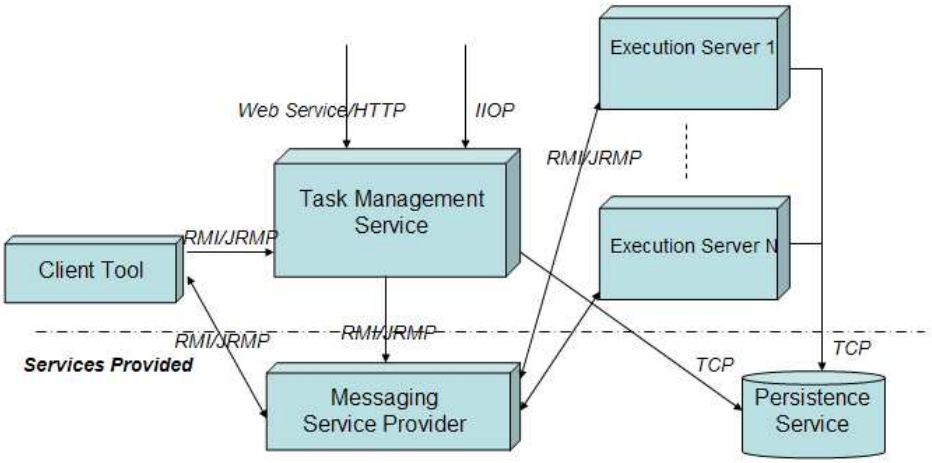


Figure 6: Deployment as Campus Grid Scheduler

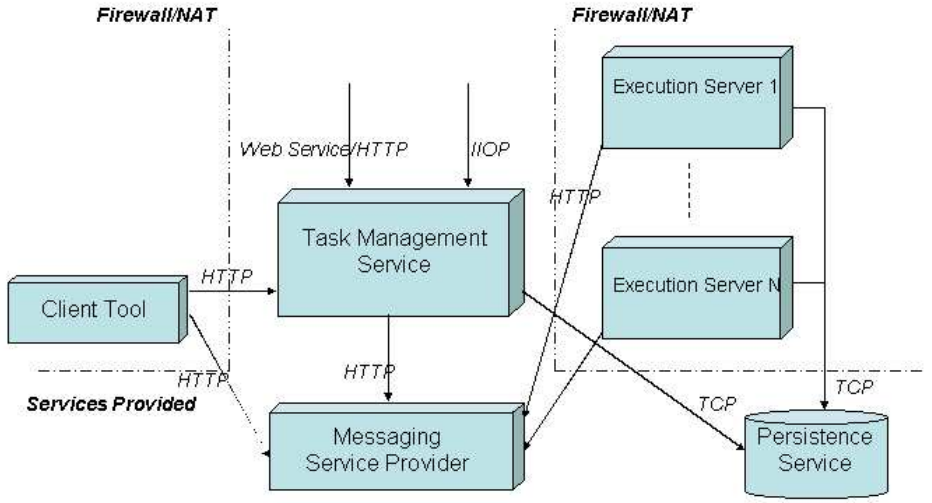


Figure 7: WAN Deployment

6 Evaluation

6.1 Functional Evaluation

We evaluate each element of the architecture to see how its scalability and robustness properties:

- Task Management Service: This service is stateless and therefore can easily be made highly available through standard load-balancing and clustering techniques. If it fails, the system can carry on processing jobs submitted. Only the clients currently connected to it will not be able to submit and control the workflows, and the check for failed execution servers will not be performed.
- Execution Server: The number of concurrent execution server is only limited by the maximum number of subscribers that the messaging service supports and the maximum number of connections that the persistence provider can handle. In case of failure, the job will not be lost. Once the failure detected, the task will be resubmitted to the queue.
- Messaging Service Provider: This is a service provided to our implementation. Its robustness and scalability characteristics depend on its implementation and on the database it uses for persistence.
- Persistence Service Provider: As for the previous provider, the robustness and scalability characteristics of the database vary with providers. While we used HSQL, which does not provide specific failover, scalability or high-availability features, there are many database vendors providing such capabilities.

6.2 Experimental Evaluation

We have implemented and tested the scheduler using a variety of Discovery Net bioinformatics and cheminformatics application workflows. A complete empirical evaluation of the scheduling is beyond the scope of this paper since it takes into account the characteristics of the application workflows themselves. However, in this section we provide a brief overview of the experimental setting used to test and evaluate the scheduler implementation.

The scheduler was deployed for testing over an ad-hoc and heterogeneous set of machines. The submission server was hosted on a Linux server where the persistence and messaging services were also held. The execution servers were running on a set of 15 Windows single processor desktop machines on the same organisation's network without restrictions. The scheduler sustained overnight constant submission and execution of workflows, each execution server handling a maximum of 3 concurrent executions, without apparent bottlenecks.

It has also been deployed over a cluster of 12 IBM Blades server running Linux, where the execution servers are running on a private network not accessible by the client machine.

Finally it was also deployed over WAN and networks with NAT. The client was hosted on the organisation's internal network, with only a private IP address, in the UK. The submission server was hosted in the US on a machine that had a public IP address. The execution servers were hosted on a private network in the US, although directly accessible by the submission server. Even though the impact of using tunnelling and pull mechanisms do affect the overall feel of the client application in terms of its latency when submitting, monitoring jobs and particularly visualising the results, because of increased communication overheads, it does not affect the performance of the workflow execution itself, which is the main concern.

The main potential bottlenecks needing further investigation are the behaviour of the messaging service with increasing number of subscribers, in particular execution servers, as well as growing size of workflow descriptions.

7 Comparison

The Java CoGKit [12] is a Java wrapper around the Globus toolkit and provides a range of functionalities for Grid applications including job submission. It is therefore based on native process execution, while our approach is to distribute executions of entities that run in a Java hosting environments. However the Java CogKit could be used in the scheduler proposed as a way to implement the scheduling policy for the Job queue, but not to submit the jobs directly.

The Grid Application Toolkit [9] has wrappers for Java called JavaGAT. This interface is a wrapper over the main native GAT engine which itself is trying to provide a consistent interface layer above several Grid infrastructure such as Globus, Condor and Unicore. Again, the difference of the approach relies on the submission of native process executions over the grid, instead of handling that process at a higher level and leaving the scheduling, potentially, to a natively implemented Grid or resource management service.

Proactive [6] takes a Java oriented approach by providing a library for parallel, distributed and concurrent computing interoperating with several Grid standards. While using the same approach as here, we based the engineering around Java commodity messaging and persistence services such that the robustness of the system mainly and the network protocols it uses depends on these service rather than on the implementation.

8 Conclusion

To be able to build a robust Java-based scheduler based on commodity services could enable a wider range of Grid applications to benefit from the rich framework provided by Java application server, and help to simplify their implementation. This paper presents a possible way to implement such a scheduler in this framework, as well as its deployment and some of its robustness characteristics.

9 Acknowledgements

The authors would like to thank the European Commission for funding this research through the SIMDAT project.

References

[1] S. AlSairafi, F.-S. Emmanouil, M. Ghanem, N. Giannadakis, Y. Guo, D. Kalaitzopoulos, M. Osmond, A. Rowe, J. Syed, and P. Wendel. The design of discovery net: Towards open grid services for knowledge discovery. *International Journal of High Performance Computing Applications*, 17, Aug. 2003.

[2] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. SETI@home: an experiment in public-resource computing. *Commun. ACM*, 45(11):56–61, 2002.

[3] Enterprise JavaBeans Technology. `http://java.sun.com/products/ejb`.

[4] M. Hapner, R. Burridge, R. Sharma, J. Fialli, and K. Stout. *Java Message Service*. Sun Microsystems, Inc., 901 San Antonio Road Palo Alto, CA 94303 USA, 2002.

[5] HSQL. `http://www.hsqldb.org`.

[6] F. Huet, D. Caromel, and H. E. Bal. A high performance java middleware with a real application. In *SC'2004 Conference CD*, Pittsburgh, PA, Nov. 2004. IEEE/ACM SIGARCH.

[7] JBossMQ. `http://www.jboss.com/products/messaging`.

[8] C. Lai, L. Gong, L. Koved, A. Nadalin, and R. Schemers. User authentication and authorization in the java platform. In *Proceedings of the 15th Annual Computer Security Applications Conference*, pages 285–290, Scottsdale, Arizona, Dec. 1999. IEEE Computer Society Press.

[9] E. Seidel, G. Allen, A. Merzky, and J. Nabrzyski. Gridlab—a grid application toolkit and testbed. *Future Generation Computer Systems*, 18(8):1143–1153, Oct. 2002.

[10] Shibboleth-aware portals and information environments (spie) project. http://spie.oucs.ox.ac.uk/.

[11] Sun Grid Engine. `http://gridengine.sunsource.net`.

[12] G. von Laszewski, I. T. Foster, J. Gawor, and P. Lane. A Java commodity grid kit. *Concurrency and Computation: Practice and Experience*, 13(8-9):645–662, 2001.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

The GridSite Toolbar

Shiv Kaushal and Andrew McNab

School of Physics and Astronomy, University of Manchester, Manchester UK

Abstract

We describe the GridSite toolbar, an extension for the Mozilla Firefox web browser, and its interaction with the GridSite delegation web service. A method for automatic discovery of the delegation service is also introduced. The combination of the toolbar and automatic discovery allows users to delegate credentials to a remote server in a simple and intuitive way from within the web browser. Also discussed is the toolbar's ability to enable the use of the GridHTTP protocol, also a part of the GridSite framework, in a similar way.

1. Introduction

Large production grids currently use X.509[1] certificates, GSI[2] proxy certificates and VOMS[3] proxy certificates as a means of authenticating users and delegating authority from users to remote servers. The use of proxy certificates ensures that a user's private key is never exposed but can enable remote servers to act on their behalf.

GridSite[4] is a security middleware project that adds the ability to accept client-side X.509 certificates and GSI/VOMS proxies to the Apache web server[5]. It also allows the generation of access control policies, in GACL[3] or XACML[6], to limit access to files/pages based on these credentials.

GridSite also provides several methods for enabling secure transfer, storage and location of files. One such mechanism, for transferring files over HTTP with access controlled through an initial HTTPS connection, known as GridHTTP, will be explored in the initial part of this paper.

GridSite also acts as a platform for hosting secure web services, written in any of the CGI scripting languages supported by Apache, using the added certificate handling capabilities to authenticate clients. It also comes with a method of “sandboxing” these services inside temporary pool accounts, enabling web servers to safely allow users to remotely deploy web services onto a GridSite server.

As an example web service, the GridSite package contains an implementation of the GridSite delegation service. This is a web service interface for delegation, designed to offer several advantages over the standard methods of transferring proxy certificates to remote servers. The details of the delegation service are discussed later.

A more extensive summary of the GridSite framework can be found in "The GridSite Security Framework"[7].

A common problem when using web services interfaces for applications is knowing how to locate the services. We introduce a method for enabling automatic discovery of a delegation service from within a web browser. The mechanism described could easily be applied to any other web service.

Finally, we describe how this mechanism is used in conjunction with the GridSite toolbar - an extension to the Mozilla Firefox[8] web browser - to allow users to easily locate and delegate credentials to available delegation services. We also show how the toolbar can allow users to make use of the GridHTTP protocol.

2. GridHTTP

GridHTTP is a protocol, defined within the GridSite framework that allows files to be downloaded over a standard HTTP connection, but first requires authentication of the clients via HTTPS. The aim of this protocol is to allow large (gigabyte) files to be transferred at optimal speeds while still maintaining some level of security. The approach used avoids the problem of authentication over HTTP (usually achieved via usernames and passwords) by using the certificate handling capabilities of the GridSite software as well as the access control list functionality.

In order to retrieve a file using the GridHTTP protocol, clients must connect to a web server over HTTPS and set the value of the

"Upgrade" header in the request to "GridHTTP/1.0". This is entirely in keeping with the HTTP standard so any server which does not understand the GridHTTP protocol, or does not have it enabled, will ignore the header.

If GridHTTP is enabled, the server will determine if the client should have read access to the requested files, based on their X.509, GSI proxy or VOMS proxy certificate and any defined access control lists. If access is allowed, the server will respond with a redirection to a standard HTTP location for the file and with a single-use passcode, contained in a standard HTTP cookie. The client should then present this passcode cookie when requesting the file from the HTTP location and will be granted access to download the file.

The performance benefits of the GridHTTP protocol come not only from the data stream being unencrypted (saving CPU cycles at the server and client ends) but also from the highly optimised Apache file serving routines. In comparison, while GridFTP[9] allows unencrypted data transfer it does not make use of low level system calls as in the case of GridHTTP/Apache. The performance difference between the two transfer methods (for unencrypted data streams) is evident in research carried out by R. Hughes-Jones[10].

3. The GridSite Delegation Service

The GridSite delegation service is a web service that allows users to place proxy certificates on remote servers. The WSDL-based service uses standard plain-text SOAP messages, with the authentication coming from an HTTPS connection from the client. This makes it very easy to implement the service in any of a variety of programming languages, especially when using GridSite as the web services platform.

3.1 Delegation Service Internals

The delegation procedure is initiated by a client sending a getProxyReq message to the service. The service then produces a public/private key pair and generates a certificate request from the public key. The certificate request is then sent to the client. The client then signs the request with their private key and sends it back to the server in a putProxy message. The signed request combined with the associated private key forms a valid proxy certificate.

This approach has the distinct advantage over the usual methods of delegating credentials to a remote server. The standard "grid-proxy-init" and job submission routines produce the public and private keys for the proxy certificate locally, sign the public key and then transfer the both to the remote server over the network. In the GridSite delegation service, the private key associated with the proxy certificate never leaves the remote server, adding an extra layer of security to the delegation process.

This is completely analogous to the processes involved in obtaining a standard X.509 certificate, but cast as a web service. The delegation service is treating the client as a Certificate Authority (CA) and requesting that the public key be signed by the user's private key (which acts like the CA's private key). The main difference is that both the client and the service can verify the identity of the other through trusted CAs, so there is no lengthy identification step involved.

The above description covers the functionality of the initial (version 1) delegation service interface. There is now an updated interface specification (version 1.1) that extends the functionality of the delegation service. The new functionality includes methods that allow users to check when an existing proxy certificate will expire, to renew such a certificate and to destroy the certificate.

3.2 GridSite Implementation

The GridSite implementation of the delegation service makes use of the gSOAP[11] toolkit and is written in C. Running under the GridSite environment allows the service to obtain authentication information about connecting clients directly from environment variables. It is intended as a simple illustration of how web services can be created within the GridSite framework but this implementation can also be incorporated into other web services that might require this functionality. Additionally, GridSite provides a command line client, also built using the gSOAP toolkit, in order to supply a complete solution.

The GridSite delegation service specification was created within the EGEE[12] project. As a result, there are also Java client and server implementations available within the gLite framework. Since the service is based on a WSDL definition, all of the combinations of client and server implementations interoperate without any problems.

4. Service Discovery

Traditional web services require that the client is either configured for one particular instance of the service, with a hard-coded location (as is the case with the GridSite delegation service command line client, once compiled), or that the user of the client provides the URL of the service they wish to use. This can cause problems if locations of services change or users wish to locate alternative services providing the same functionality.

A system was developed to allow a browser (and in turn the GridSite toolbar) to be notified of an available delegation service by a web site. The URL of the service is provided either in HTTP headers or in a META[13] element in the HTML source of the page being viewed. The META element method is similar to the method employed by sites and browsers to enable automatic location of RSS feeds, which uses a LINK element.

The header used for the notification is "Proxy-Delegation-Service", the value of which should be the URL of the delegation service. The insertion of this header can be easily achieved with GridSite Apache module and setting the GridSiteDelegationURI directive in the web server's configuration file.

To deliver the same information without using HTTP headers, a META element containing the http-equiv attribute can be inserted into the HEAD element of a web page. The META element would have the following format:

```
<META
http-equiv =
"Proxy-Delegation-Service"
content =
"https://eg.com/delegate.cgi"
>
```

The combination of the two methods above provides great flexibility for different groups of people to alert users to a delegation service. The HTTP header method allows a web server administrator to inform all visitors of a related delegation service and META element method allows an individual page (a personal home page, for example) to do the same. The latter may be useful in an environment where enabling server-wide options are not possible (e.g. pages located on unrelated servers) or on a server not using the GridSite software.

5. The GridSite Toolbar

As stated previously, it is possible for users to upload custom web services to a GridSite server, an example of which is the delegation service. Additionally, GridSite provides a wide variety of site management features, including editing/uploading/deleting files and folders and editing access control lists. All of these tasks can be carried out from within a web browser.

Although a command line client was created for the delegation service, it did not provide an easily accessible interface as in the case of the browser enabled functionality mentioned above. A browser based client for the delegation service was produced to illustrate a mechanism for allowing interaction with such web services.

The GridSite toolbar is a client for both the GridSite delegation service and the GridHTTP protocol, wrapped up in an extension for the Mozilla Firefox web browser. It makes use of several features of the browser and the service discovery method, described in Section 4, to make delegation and using GridHTTP a simple point-and-click task.

5.1 Mozilla Firefox

Mozilla Firefox was chosen as a base platform for the GridSite toolbar for several key features. These include:

- Default web browser in the Scientific Linux distribution, recommended by EGEE/gLite.
- Explicitly designed to be easily extensible.
- Provides JavaScript objects for manipulation of SOAP messages and interacting with WSDL services.
- Provides a cross-platform development environment.
- Simple API for creating graphical interface elements.
- Built in certificate verification of remote servers over all HTTPS connections.
- Security updates come "for free" from Mozilla.

In addition to these features, the use of Firefox keeps to the GridSite project's general philosophy of building on established, open source software and related protocols, such as Apache and HTTP, as much as possible. These projects have large development teams and are extensively tested by a wide user base. This allows the GridSite software to inherit the stability, performance and security of these projects and receive security updates for crucial elements (such as HTTPS communication – client and server side) for free. Trying to create stable, efficient solutions for the functions that GridSite and the toolbar provide in the form of bespoke software and protocols would be much more time consuming and much harder to maintain against possible security flaws.

5.2 Requirements

The GridSite toolbar makes several assumptions about the configuration of a user's environment. It requires that the user's certificate (as well as the relevant CA root certificate) is loaded into the Firefox software security device. It is also assumed that the user's certificate is stored in PEM encoded usercert.pem and userkey.pem files in the user's ~/.globus directory. Finally,

the extension also requires OpenSSL[14] command line tools to be installed.

The toolbar has been developed to work on Linux systems but could be ported to work in a Windows environment, provided that a method of producing secure named pipes, or an equivalent, is available (see section 5.4 for details of how these are used).

5.3 Service Detection

Every time a page (or tab) is loaded or brought into focus, the HTML source of the displayed page is searched for the relevant META element. Then an HTTP HEAD request is made to the same URL as the page being displayed and the response is inspected for the Proxy-Delegation-Service header. If location is defined in both the HTTP headers and the META element the value found in the headers will be used.

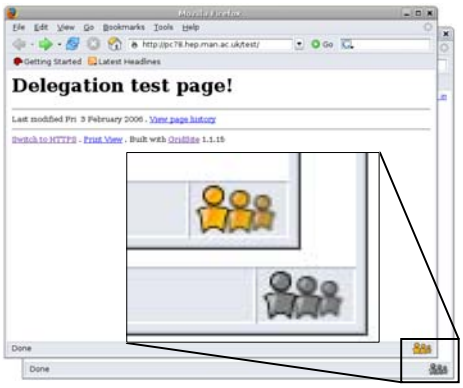


Figure 1: The two states of the delegation service detection status bar icon

The detection of a delegation service is indicated by an icon in the status bar of the browser window (as shown in Figure 1). When a service is detected, the icon is "lit up" and becomes active. In this state, clicking it will initiate the delegation procedure. When no delegation service is detected, the icon will become greyed out and clicking it will produce a message informing the user that no service was found.

5.4 Delegation Procedure

The delegation procedure involves several steps. Firstly a getProxyReq SOAP message is constructed and sent to the discovered URL. The service responds with a certificate request, which is saved to disk in a temporary location.

This step (and every subsequent communication with the delegation service) is carried out using Firefox's HTTPS capabilities. The browser's built in functionality will produce a warning to the user if the server hosting the delegation service has a certificate that is either invalid or not from a recognised CA. The connection also uses the certificate in the software security device to authenticate to the service.

The user is then asked for their PEM passphrase, which is passed, through a named pipe, to a shell script wrapped around the OpenSSL command line program. This script signs the request to produce the proxy certificate. The shell script makes use of a file called usercert.srl in the user's ~/.globus directory (from the -CAcreateserial command line option for OpenSSL X.509 tools) to ensure that the extension will never produce two proxy certificates with the same serial number – as recommended in the IETF RFC3280 specification for GSI proxies.

Finally, the certificate is read in, inserted into a putProxy SOAP message and sent to the delegation service.

5.5 User's Experience of Delegation

The majority of the details described above are hidden from the user when using the toolbar. Upon clicking the delegation button in the status bar, a short dialogue box appears informing the user that the browser is attempting to connect to the delegation service.

The user may then be prompted for up to 2 passwords. The first is for the Firefox software security device and will only be asked once per Firefox session. This is the standard Firefox behaviour when using the user's certificate to authenticate to a secure server. The second password requested is the PEM passphrase for the ~/.globus/userkey.pem file. For security purposes, this passphrase is requested every time the GridSite toolbar is used to delegate to a service.

Once the proxy request has been signed and sent back to the server, a confirmation dialogue box will appear.

5.6 Limitations

There are some limitations in the current version of the GridSite toolbar with regards to the interaction with the GridSite delegation service. Firstly, proxy certificates can only be created with validity times in multiples of 24 hours. This is due to restrictions in the OpenSSL command line tool. The time is currently locked at 24 hours, but there is room to allow for varying lifetimes.

Additionally, there is currently no method offered to allow users to select a service from

META tags in preference over a service specified in HTTP headers. This can be achieved through the use of directory specific Apache settings by disabling the HTTP header for a particular area of a site.

Finally, the new features introduced in version 1.1 of the delegation service interface are not yet supported by the GridSite toolbar. This could be achieved relatively easily by extending the work already done.

5.7 GridHTTP

Access to files using the GridHTTP protocol is similarly an easy operation when using the GridSite toolbar. To do so, users right click a HTTPS link to a file and select "Get with GridHTTP" (Figure 2).

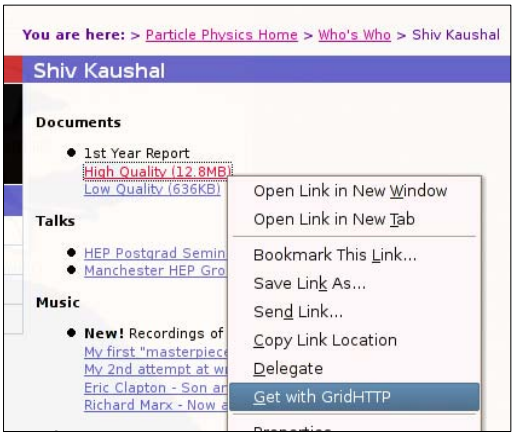


Figure 2: Using GridHTTP to download files from within Firefox.

The extension then sets the location of the current window to the right-clicked URL and intercepts the request to add the required Upgrade header. After this point no further intervention is required from the toolbar. The default Firefox behaviour is to follow the redirection from the server and to present the passcode cookie. Upon doing this the browser will handle the file as normal – either displaying it within the browser window or prompting the user to save or open the file with an external application.

6. Conclusion

The GridSite toolbar is a simple example of how the GridSite functionality can be combined with the service discovery method and a web browser environment in order to simplify the use of web services. It not only avoids the use of long command line strings, but also uses the built in functionality of the web browser (such as choosing where to save a file) to add graphical elements that make the process more intuitive.

Using the web service hosting features of GridSite and the methods outlined in this paper, it is possible for any user to create and deploy a web service, or a collection of services, configured to their own specific set of requirements. The web service can be written in almost any programming or scripting language, hosted as CGI scripts for Apache, and then made accessible through the familiar environment of a web browser.

Applications could range from something as simple as a having a mechanism for notifying users of the status of submitted jobs to a complex Grid "portal" site, which could be used to submit jobs and retrieve output. In the latter case, the GridSite delegation service and GridSite toolbar functionality could be easily integrated to allow the delegation of credentials to the service as required.

The GridSite toolbar demonstrates that making grid applications as easy to use as any locally installed application is possible. Using such techniques can help make the Grid more accessible for new users and ease its adoption.

Acknowledgements

This work was funded by the Particle Physics and Astronomy Research Council through their GridPP end e-Science Studentship programmes.

We would also like to thank other members of the EDG and EGEE security working groups for providing much of the wider environment into which this work fits.

References

1. X.509v3 is described in IETF RFC2459, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile."
2. Grid Security Infrastructure information is available from Globus: http://www.globus.org/toolkit/docs/4.0/security/
3. VOMS and GACL are described in the EDG Security Co-ordinations Group Paper, "Authentication and Authorization Mechanisms for Multi-Domain Grid Environments", L. A. Cornwall et al, Journal of Grid Computing (2004) 2: 301-311.
4. GridSite Software is available from http://www.gridsite.org/
5. The Apache Web Server: http://httpd.apache.org/
6. XACML specification is by OASIS: http://www.oasis-open.org
7. "The GridSite Security Framework", A. McNab & S. Kaushal, Proceedings of All Hands Meeting (2005).
8. Mozilla Firefox information and downloads: http://www.mozilla.com
9. GridFTP: http://www.globus.org/grid_software/data/gridftp.php
10. Richard Hughes-Jones, private communication and talk at GNEW 2004.
11. The gSOAP C++ web services toolkit is available from http://www.cs.fsu.edu/~engelen/soap.html
12. The EGEE (Enabling Grids for E-sciencE) project: http://public.eu-egee.org/
13. HTML 4.01 Specification detailing the use of all valid elements: http://www.w3.org/TR/REC-html40/
14. OpenSSL is available from http://www.openssl.org/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Metadata-based Discovery: Experience in Crystallography

Monica Duke

UKOLN, University of Bath

Abstract

Facilitating discovery is an aspect of curation that has been addressed by the eBank UK project. This paper describes the metadata and associated issues considered when working within the chemistry sub-discipline of crystallography. A metadata-mediated discovery model has been implemented to aid the dissemination and sharing of research datasets; although the specific characteristics of crystallography were the main driver in determining the metadata requirements, consideration was also given to cross-disciplinary interaction and exchange. An overview of the metadata profile that has been developed is provided, against the background of the project.

1. Discovery in the curation life-cycle.

Digital curation can be viewed as the active management of data over the life-cycle of scholarly and scientific interest, and is the key to reproducibility and re-use. [1] Within this view, curation is seen as encompassing activities that support the *immediate* use of data as well as its long-term preservation.

Metadata for resource discovery and retrieval is considered to play an important role in this process. Metadata-mediated discovery relies on the description of the data in such a manner as to support discovery services that match a search requirement against some characteristic of the data. Metadata is intended to promote contemporary discovery and use, as well as future unintended uses.

The eBank UK project has addressed a perceived shortfall in the current publication and dissemination process in the field of crystallography, by designing and implementing an open access repository and improving dissemination routes for the associated metadata.

1.1 Sharing and Discovering Crystallographic Data

McMahon [2] provides a historical overview of electronic metadata management in the field of crystallography, from the publisher's perspective. After referring to the processes around bibliographic metadata management in scientific journals, (which are common across publishers working in an electronic environment), the overview deals with the scientific metadata in crystallography, and particularly the standard data exchange format CIF [3], which the community developed in the nineties. The paper highlights the tradition of depositing data in support of published articles in the field of crystallography, and the success achieved by policies to archive data in the agreed uniform format. The CIF format is itself rich in metadata characterizing the data and results derived from it.

A number of repositories of crystallography data are in existence or in development with the aim of allowing human users to query, discover and access their content. They vary in their subscription access model (free, partial or fully fee-based), and in the range of additional services offered e.g. visualization software. (for two example repositories see [4] and [5]). What is common is that all these repositories restrict searching capabilities to the entry of search criteria on a web form as the sole point of discovery, (with an additional email request required to access the data in some cases).

Whilst [2] suggests a number of potential ways of sharing queries across these repositories, and points to example technologies that may fulfil that role in the future, it is clear that an integrated method of discovery across crystallography resources is not yet a reality, despite the use of the common CIF format for storing results data.

1.2 The contribution of the eBank UK project.

The eBank UK approach to data sharing has been two-pronged: the provision of a human-browsable repository of data, and the exposing of descriptive metadata using an internationally agreed protocol, the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [6].

The details of the archive and the chosen protocol are described in other contributions submitted to this workshop, or elsewhere [7], [8], [9], [10] and shall not be repeated here. Briefly however, datasets created during a

crystallographic determination are deposited into an archive. During deposition, a number of metadata items are captured, either by automatic extraction from the data or through input by the depositor. The data is then made available through a web interface and can be searched or browsed. Each entry displays the key metadata and gives access to the underlying data. Additionally, the metadata can be accessed through a machine interface using a harvesting model. Third parties are able to issue http requests to retrieve the XML-formatted metadata, which can then be stored locally, integrated and cross-queried with any other metadata available.

The advantage of this approach is that it opens up the metadata to be re-used and incorporated into multiple services, thus potentially generating new pathways to the data. It is not only human users who can query, discover and access the content. The requests for metadata need not be initiated by a human user but can be delegated to an automated process. A machine-processable format of the metadata is provided to promote flexibility in its use and diversity in the discovery services that can be layered above. In particular, one focus of the project was to explore the link between digital library technologies (such as the OAI-PMH) and the sharing of scientific data and metadata, and potential connections with the published literature as a method of accessing data.

1.3 Other Initiatives

The eBank UK project is of course not unique in offering metadata as a discovery tool for a data-rich repository. Examples from eScience initiatives include the NERC DataGrid, which developed an extensive metadata model, part of which was directed at discovery services [11].

The FEARLUS-G project designed an OWL ontology [12] that is used in a service that allows land-use scientists to access and re-use results and observations.

AstroGrid is part of the International Virtual Observatory Alliance collaboration, working towards interoperability of astronomical data. Standardisation of metadata has been one of their efforts, and a number of schema have been developed, including a recommendation on resource metadata.

The MyGrid Ontology supports a large number of components that deals not only with data but also workflows, people, specifications, organizations and services.

2. The eBank UK metadata application profile

The definition of a metadata profile to describe the datasets made available for discovery in this project has been an ongoing activity. Agreement has been reached gradually and the profile has evolved and been refined over time, through a number of iterations. The current product reflects the input of a multidisciplinary team with the digital library influence making a definite impact. Knowledge of the specific application area was provided directly by the crystallographers, all active practitioners in their field, who were involved throughout. Additionally, two workshops as well as desk research were used to gain a wider perspective of other scientific areas and to validate the outputs with community leaders.

2.1 Starting from Dublin Core

The metadata that is exposed from the dataset archive via OAI-PMH is intended to fulfil a dissemination role, leading to a number of discovery services. The Dublin Core standard has since its inception been intended as a resource discovery standard and carries support in the digital library communities. It was thus evaluated first and has now formed the basis of the metadata exposed by the crystallography metadata service.

The Dublin Core consists of an Abstract Model (of resources being described by metadata descriptions), a number of metadata terms used to describe resources, encoding guidelines (for XML and RDF), and schemas (in different schema languages) defining the Dublin Core terms. It should be noted that the abstract model was still in development when the eBank project started; it was finalised in March 2005. The eBank UK project has specified

- a number of Dublin Core metadata terms that can be used in the description of a crystallography data resource,
- their encoding in XML, and
- an XML schema definition for metadata exchange.

Furthermore, internationally-agreed guidelines exist for documenting the use of Dublin Core in a specific application (a so-called Application Profile) and documentation according to these guidelines is provided at [13]. The aim of this documentation is to assist others either wishing to create instances similar to the eBank repository, or to process the

metadata exposed by the repository, thus facilitating re-use.

2.2 Qualified Dublin Core

Dublin Core is presented as two levels. The simplest and most basic level consists of a basic element set. This is then extended to allow for element refinements, that is elements that refine (but not extend) semantics in ways that may be useful for resource discovery.

Two categories of refinements (also called qualifiers) are recognised; the first category makes an element narrower or more specific, e.g. the *relation* element can be refined to isVersionOf or isFormatOf. These are two specific instances of the relation element which narrow down the meaning. The second method of qualifying an element is by specifying an "encoding scheme" which aids in the interpretation of the element value. Examples of encoding schemes include controlled vocabularies or formal notations. Thus an element qualified by the encoding scheme will have as its value a token taken from a controlled vocabulary (e.g. a classification system) or a string formatted according to a formal notation e.g. expression of a date.

The eBank UK application profile makes use of the basic Dublin Core elements as well as the two mechanisms of refinements to define an element set that is specialized to the domain-specific needs. The 'dumbing-down' principle of Dublin Core ensures that applications that cannot recognise the specialised element can safely ignore the qualifier and treat the element as unqualified. Despite the loss of specifity this is intended to allow the remaining value of the element to be used correctly and usefully for resource discovery.

2.3 Generic Dublin Core descriptions

The Dublin Core metadata elements (indicated in this section by the prefix dc) comprise a number of elements that can (intentionally) be applied to a large number of different resources. For example dc:title is the name given to a resource. This label can be applied equally to the title of a book, a photograph, a work of art or an experiment. In the eBank application profile, the title of the datasets takes the name of the molecular compound that is being determined in the crystallographic experiment.

dc:date is a date associated with an event in the life cycle of a resource. Typically the date is associated with the creation or availability of the resource; the date was considered a useful element to include since it could allow users to limit searches for datasets added within a specific time period. A scientist (or service) could then generate a 'latest additions' or 'added since' feature.

dc:creator is the entity responsible for making the content of the resource. This has been interpreted as the names of the scientists themselves who create the datasets that are deposited. Previous work has identified issues of metadata quality and it is recommended that content rules should be applied to the values of metadata elements to improve quality [14]. One recommendation is that author names should be entered in controlled form and this has been implemented in the deposition software, so that name formats adhere to those recommended for eprints [15]. The organisations that the scientists belonged to was also included but this was designated within the dc:publisher element, which is defined as an entity responsible for making the resource available.

dc:type describes the nature or genre of the content of the resource. Various types of resources are described and disseminated using OAI-PMH (text, image, audio and video are but a few examples available from OAIster [16]), therefore the value of this element was considered important to enable selective and sorting operations on the harvested metadata in a heterogenous environment. The value currently implemented for the content of this element is 'crystal structure data holding'. There are other potential values that this element could take (not currently implemented), such as the more generic 'dataset'. Note that all elements in Dublin Core are repeatable therefore using all the values in repeated dc:type elements is an option.

dc:identifier is defined in the Dublin Core documentation as an unambiguous reference to the resource within a given context. In the context of the service being deployed in the eBank project, the most useful identifier would be one that not only identifies the datasets but can also be dereferenced to access the resource.

Two types of identifier are being used. The URL of the web entry for a holding in the archive is used as it provides an entry point to all the datasets for a crystal structure. On clicking the URL, the web page for the entry displays links that enable download of various datasets, and a selection of key metadata about the resource.

An alternative identifier is also being implemented. This is the Digital Object Identifier (DOI). DOIs are assigned in collaboration with a German agency. The DOI

system of identification includes a network of registration agencies. Alongside the management of identifier assignment, agencies also record metadata about resources registered. It is hoped that besides the technical details and infrastructure, this more formal approach to identifier assignment will inject a degree of commitment towards, and permanence of, the resources registered, thus ensuring persistence of identifiers and their resolution. This is a very relevant issue to harvesters of the metadata since unreliable identifiers would affect the quality of the discovery services provided using harvested metadata. The choice of DOI was partly influenced by collaboration with publishers in the crystallography literature, (described further in section 3.1), since the DOI has been especially successful in uptake with publishers.

dc:isReferencedBy is a refinement of dc:relation that is intended to contain pointers to published literature that specifically refer to the data in the archive. This element may contain either a textual citation to the publication, or an actionable URL, or other identifier.

dc:rights provides information about rights held in and over a resource. This is crucial information both for the data provider who will want to assert the rights held and granted over a resource that is being made available, and dually to a potential user who has discovered the resource. Discussion with third parties planning to use the harvested metadata indicated that this information would be absolutely necessary for them to determine what sort of access and use of the resources was allowed, thus determining what discovery services could be provided. In the eBank implementation the rights are described by means of a statement declared on a web page, the URL for the page is then exposed in the dc:rights metadata element. This is possible since all the resources in the repository have been given the same right of use by the depositors.

2.4 Chemistry-based descriptions

Crystallographic determinations involve the analysis of chemical structure and the resulting datasets often consist of three-dimensional co-ordinates. However, for the purposes of naming the structure, more condensed forms (textual and/or formulaic) are used. These can all be of assistance to the crystallography researcher in identifying molecules of interest and are thus important metadata to include in a search service.

One example of how chemical names have been included in the eBank data is the use of the International Union of Pure and Applied Chemistry (IUPAC) name in the title. The generation of this title requires the use of guidelines and the contribution of human expertise as well as software processing.

Additionally, a number of other chemistry metadata is provided in multiple dc:subject metadata elements. dc:Subject describes the topic of a resource and since the chemical structure is the main topic of the datasets, the different ways of identifying the structure have been included. Since the different naming mechanisms have different rules for how they can be formulated and formatted, a typical search service would need knowledge of the formats present in the metadata, so that appropriate search criteria can be given to the user. The Dublin Core encoding schemes refinement mechanism has been applied to distinguish between the different naming conventions. Thus chemical formulae are considered to be taken from the controlled vocabulary that consists of all chemical formulae that can be expressed. Similarly, InChIs (a recently agreed international standard for uniquely identifying molecules) are considered to be another controlled vocabulary which can be applied to dc:subject to restrict the values which that element can take when this encoding scheme is declared.

Another form of controlled vocabulary has also been defined by the eBank project. This consists of a small set of terms which assign the dataset to one of four types of compound class (organic, inorganic, bio-organic, organometallic). Once again this is a specialised type of vocabulary that was considered useful in the application domain to enable users to narrow down searches to their field of interest when carrying out discovery activities.

One further use of the encoding scheme mechanism provided by Dublin Core was made to define a list of types for datasets. The values of dc:type in eBank can be further refined by declaring the type to be one of the eBankDatasetTypes. These consist of a closed list (though extension in the future is possible) and are declared in the published XML schemas.

2.5 Exposing Complexity

One aspect of the repository that has been glossed over so far is the granularity of the resources provided. Each crystal structure determination consists of a number of experimental stages, with datasets being produced at each stage. The datasets may vary,

from images of x-ray diffraction patterns, to structured documents, such as the CIF. One aim of the repository was to improve on the current publication and dissemination processes (which make the CIF file available) by giving access to **all** the results from a determination. Each entry in the repository therefore consists of a collection of files from different stages of the experiment, all arising from the determination of one chemical structure.

From the discovery perspective, the metadata should reveal to the user the existence of the multiple files available, together with an indication of their nature. In this specific domain, the experimental stage from which the files are generated is considered significant and indicative of the files available. Altogether, the files make up a collection, or more specifically a data holding, and share common characteristics such as the creator, and chemical descriptors.

The model of the crystallography resource that emerges bears some resemblance to a category of digital objects termed *complex objects*. In the digital library domain, content packaging standards are being proposed as a tool to disseminate digital resources that are composed of more than one underlying file or object. Typically these standards consist of a mechanism for declaring the structure and make-up of digital items, and use XML schemas that define an XML format that either references or contains the data files that make up the package.

METS [17] is one such packaging format being investigated by the eBank UK project. METS is maintained by the Library of Congress; a METS document consists of seven major sections, including a METS header and descriptive metadata. The descriptive metadata section is designed so that metadata descriptions (e.g. those in Dublin Core) can either be included or referenced. This allowed the project to re-utilise the Dublin Core descriptions, outlined above, within a METS file.

The File section and the Structural Map section of METS are used to link (or contain) the files, and to describe the logical or physical relations, respectively. These are the sections currently being implemented and investigated.

The Behaviour section of METS (in common with other sections of some of the other packaging formats) associates files with executable code needed to read or manipulate them. The eBank project has so far concentrated on the dissemination role of the metadata, simply advertising the existence of datasets; specialised sources already exist with integrated facilities for access coupled with data manipulation, and it was not the intention of eBank to compete with these sources. We therefore do not have any immediate plans to implement this feature of the METS standard. However it is an obvious advantage that the standard accommodates this information which is available for others to use should they wish to extend on the eBank output.

3. The Realities of Cross-Discovery.

As a cross-disciplinary effort, one of the interests of the eBank UK project was to encourage and explore common technologies across disciplines and resource types, with a focus on linking between data sets and published literature. Given its uptake in the digital library world, and the experience of the project partners, the OAI-PMH was viewed as a promising candidate protocol to provide shared technology connecting digital libraries with scientific repositories.

3.1 Connections with published literature

Initially it was envisaged that an OAI-PMH repository of crystallography literature with known connections to the deposited data would be created for initial demonstration of cross-search capabilities. Unlike some other disciplines, (such as physics), there was no existing body of publications of crystallography that was already accessible in this way. This plan was superceded since good collaborations that developed between project partners and the IUCr resulted in publication metadata being made available by the latter body for the purposes of demonstration in a web interface. Thus XML descriptions of a small sample of carefully selected publications were cross-searched with the metadata from the data archive.

The searching capabilities were shown in a demonstrator prototype which supported searching against author/creator names and chemical information. The publications included had been identified such that connections were known to exist between the articles and the datasets.

The use of the OAI-PMH had an unplanned consequence. Discussions are in an advanced stage to allow publishers to use the protocol to access data that is usually submitted in support of published articles. The use of the DOI also means that rather than reproduce the data in the published article (as sometimes happens), the data will simply be referenced with more space in the article devoted to discussion of the

results. Thus although not exactly as originally intended, OAI-PMH looks set to become part of the infrastructure for publishing data in crystallography, and this may lead to new dissemination routes for the data.

3.2 Connections with eLearning

One other angle that the eBank project is exploring is the pedagogical role of data. A study is currently being undertaken to examine the interaction of students directly through the web interface of the archive containing the data. However, from the discovery perspective, the potential of the metadata to be used to identify relationships or relevance between datasets and e-learning materials from other sources is still largely unknown.

The OAI-PMH is being used in a growing repository of learning resources funded by the JISC. JORUM [18] is a free online service for teaching and support staff in UK Further and Higher Education Institutions, helping to build a community for the sharing, reuse and repurposing of learning and teaching materials. It is hoped that searches can identify resources from JORUM that complement the datasets in the eBank repository, when used by students to search for data.

With such a specialist area as that addressed by eBank to date, it is always questionable whether the coverage of a generic resource such as JORUM will be wide enough to contain a sample of results relevant to an eBank search. Initial examination of the metadata revealed that a potential subset of resources (albeit small in number) were relevant to the topic of crystallography. Regretfully, at the time of writing access to the learning resources themselves was still being negotiated (institutional sign-up and ATHENS authentication is required). Therefore their actual relevance and the target audience could not be evaluated by the crystallography users in a demonstrator prototype. It is hoped that once the access issues are sorted out a demonstration and evaluation can take place.

3.3 Validating the output

Ideally, the true test of the metadata profile would be demonstrated bv a cross-search service against a number of different repositories containing either crystallographic, or indeed other resources. Due to the lack of suitable compatible repositories (i.e. ones supporting OAI-PMH) it is still too early to be able to carry out such validation.

3.4 Use of Dublin Core and OAI-PMH in eScience

[19] in a report on data curation for eScience in the UK in 2003 found an "almost total lack of knowledge of metadata tools such as the Dublin Core" in the responses to a questionnaire. However eScience initiatives (such as those quoted in Section 1.4) do report using some of the Dublin Core elements amongst a number of other specialized descriptions adapted to their applications. There is not much evidence however of use of the OAI-PMH as a common framework to deliver research data or metadata (at least in the UK) to date.

4. Future Work

The eBank UK project has defined the metadata fields (described in sections 2.3 and 2.4) based on typical laboratory practice at a large national centre, with the aim of providing an example that could be re-used in other settings. Furthermore, the software that generates the metadata is specialised to the workflow at that laboratory. Work is ongoing at other centres to evaluate alternative OAI-PMH software in different laboratories, and it is hoped to evaluate the eBank software modifications against different laboratory practices in the future.

So far only a relatively small sample of datasets have become accessible and searchable through the eBank initiative. It is therefore difficult to evaluate the efficiency of searching the metadata fields exposed until a critical mass of metadata is available. Furthermore, for some of the fields used (e.g. the InChI), experience is still being gained within the discipline to establish helpful searching methods (e.g. substructure searching).

However the notion of open access for crystallography data has been widely disseminated within the crystallography community, and generally very well received. Discussions are ongoing and funding is being sought to enable co-operation with other existing repositories to expose their metadata using OAI-PMH. These efforts together with existing activities may in the future allow for a more extensive evaluation of cross-searching and related discovery issues, with feedback from a larger group of users, beyond proof of concept.

The inclusion of a DOI as a citation for a dataset when referenced in a publication is still in the very early stages (although at least some publishers are receptive to the idea). This practice will be tested in the near future so that the reaction and interest of the community can

be gauged. Other quantitative measures of success will include actual access and download of the datasets as a result of the links appearing in the literature.

The present architecture of the eBank UK project centres on the OAI-PMH. However it is relatively straightforward to support other protocols for searching the harvested metadata, for example the demonstrator search service uses Z39.50 software to support its indexing and searching functionality. It would be possible to expose the metadata to outside parties using that protocol or related ones (such as the SRU [20]) if the use case was made.

5. Conclusion

As increasing amounts of data are generated and made available for sharing and re-use through eScience initiatives, the curation responsibility of enabling discovery of such data will become a more pressing issue. The eBank UK project has demonstrated the use of the OAI-PMH as a technology for metadata-mediated discovery in crystallography. For this protocol to become the connecting technology between repositories of crystallography data, and provide a shared infrastructure across different resource types, a more widespread adoption of the protocol amongst the applicable repositories would be required.

6. Acknowledgment

The eBank UK project http://www.ukoln.ac.uk/projects/ebank-uk/ is funded under the JISC Semantic Grid and Autonomous Computing Programme. The project partners are UKOLN, University of Bath, School of Electronics and Computer Science, University of Southampton, The School of Chemistry, University of Southampton and PSIgate, University of Manchester. The work reported here is a result of collaborative effort between project members, (past and present) from these institutions.

7. References

[1] Rusbridge C., et al *The Digital Curation Centre: A vision for Digital Curation.* From Local to Global: Data Interoperability — Challenges and Technologies, Mass Storage and Systems Technology Committee of the IEEE Computer Society, 20–24 June 2005, Sardinia, Italy

[2] Mc Mahon, B. *Semantically Rich Metadata in Crystallographic Publishing.* EUNIS 2005, University of Manchester, UK.

[3] International Union of Crystallography. http://www.iucr.org/

[4] Crystallography Open Database. http://www.crystallography.net/

[5] RCSB Protein Data Bank. http://www.rcsb.org/pdb

[6] The Open Archive Initiative Protocol for Metadata Harvesting. http://www.openarchives.org/

[7] Duke, M., Day, M. ,Heery, R. et al. *Enhancing access to research data: the challenge of crystallography.* In: Proceedings of the 5th ACM/IEEE Joint Conference on Digital Libraries, Denver, CO., USA, June 7-11, 2005. New York: Association for Computing Machinery, 2005, pp. 46-55. ISBN 1-58113-876-8.

[8] Heery, R., Duke, M., Day, M. et al. *Integrating research data into the publication workflow: eBank experience* In: Proceedings PV-2004: Ensuring the Long-Term Preservation and Adding Value to the Scientific and Technical Data, 5-7 October 2004, ESA/ESRIN, Frascati, Italy, Noordwijk: European Space Agency, 2004, pp. 135-142.

[9] Coles, S., Frey, J., Hursthouse, M. et al. *Enabling the reusability of scientific data: Experiences with designing an open access infrastructure for sharing datasets.* Designing for Usability in e-Science. International Workshop, NeSC, Edinburgh, Scotland, 26-27 January, 2006

[10] Coles, S.J., Frey, J.G., Hursthouse, M.B. et al. *An e-Science environment for service crystallography - from submission to dissemination.* Journal of Chemical Information and Modeling, Special Issue on eScience. 2006 (In Press).

[11] O'Neill, K., Cramer, R., Gutierrez, M., et al. *A specialised metadata approach to discovery and use of data in the NERC DataGrid.* Proceedings of the U.K. e-science All Hands Meeting, 2004.

[12] Pignotti, E., Edwards, P., Preece, A., et al. *Providing Ontology Support for Social*

Simulation. Proceedings of the First International Conference on eSocial Science, NCeSS/ESRC, Manchester, 2005.

[13] The application profile for crystallography data. http://www.ukoln.ac.uk/projects/ebank/schemas/profile

[14] Guy, M., Powell, A and Day, M. *Improving the Quality of Metadata in Eprint Archives* Ariadne, Issue 38, January 2004. http://www.ariadne.ac.uk/issue38/guy/

[15] Powell, A., Day, M and Cliff, P. *Using simple Dublin Core to describe eprints.*, March 2003 http://www.rdn.ac.uk/projects/eprints-uk/docs/simpledc-guidelines/

[16] OAIster. http://oaister.umdl.umich.edu/

[17] Metadata Encoding and Transmission Standard. http://www.loc.gov/standards/mets/

[18] JORUM. http:///www.jorum.ac.uk

[19] Lord, P. and McDonald, A. eScience Curation Report, JISC, 2003

[20] Search/Retrieval via URL http://www.loc.gov/standards/sru/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Curation of Chemistry from Laboratory to Publication

"*The curation of laboratory experimental data as part of the overall data lifecycle*"

Simon Coles, **Jeremy Frey**, Andrew Milsted

School of Chemistry, University of Southampton, Southampton, SO17 1BJ, UK

Abstract

The paper will illustrate the "CombeChem Project" experience of supporting the chemical data lifecycle, from inception in the laboratory to organization of the data from the chemical literature. The paper will follow the different parts of the data lifecycle, beginning with a discussion of how the laboratory data could (or should) be recorded, and enriched with appropriate metadata, so as to ensure that curated data can be understood within its original context when subsequently accessed, as it is generated (the ideal of "Autonomic Annotation@Source"). Intrinsic to our argument is the recording of the context as well as the data, and maintaining access to the data in the most flexible form for potential future re-use for purposes that are not recognised when the data was collected. This is likely to involve many routes to dissemination, with data and ideas being treated by parallel but linked methods, which will influence traditional approaches to publication and dissemination, giving rise to a Grid style access to the information working across several administrative domains summarized by the concept of "Publication@Source".

1. Introduction

e-Science[1] is about global collaboration in key areas of science and the next generation of infrastructure that will enable it. It involves the "end-to-end" linking of data and information in the face of the data deluge created by emerging experimental techniques. CombeChem[2], an EPSRC e-Science pilot project, took this vision as its focus and involved a significant number of collaborators, spread over several disciplines, based in multiple departments at Southampton, together with several other academic and industrial concerns. The project concentrated on Grid-enabled chemistry, involving synthetic, laser and surface chemistry, and crystallography, as examples of the development of an e-Lab, using pervasive computing technology to record information on all aspects of laboratory work and carry this information forward through the whole chain of generation of chemical knowledge.

We aimed to provide the digital support for an end-to-end knowledge sequence in which an experiment produces data, from which results are derived, then searched for patterns, from which conclusions are drawn, leading to further experiments. The progress of science depends on each scientist building on the results produced by others; re-use of data, in both anticipated and unanticipated ways, is vital. E-Science techniques, as demonstrated by CombeChem, enable more data to be more freely available to scientists worldwide from heterogeneous sources, a problem with which industry has wrestled for years. CombeChem has successfully addressed these problems in both practical and theoretical ways. The scientist is crucial and thus the emphasis on usability.

At first glance it might seem that our applications are domain specific but the approach taken has produced generic applications and demonstrated that the software developed for one application can be re-used in another context, i.e., in university, and even in secondary (cf e-Malaria)[3], education. In the interests of widespread applicability and use of our systems, we have made creative use of the Web and developing Grid, and of open source and free software, which has enabled us to deliver data and user-friendly tools, and we have consistently emphasised the importance of standards (e.g., InChI[4]) and have been early adopters of such standards. Over its lifetime, the CombeChem project consistently increased its international visibility and reputation. Building on established expertise in Grid computing at IT Innovation, CombeChem led the world in adopting Web Services as its base platform from the earliest point in the programme, a position subsequently adopted by the wider UK e-Science community.

CombeChem was the basis of the original Semantic Grid report[5] and showed how to achieve full integration of laboratories and

experimenters into an e-Science infrastructure based on pervasive and Semantic Grid technology. Many phases of the knowledge cycle were explored in CombeChem, from user interaction with Grid-enabled high-throughput analysis, fed by smart laboratories (notebooks and monitoring), together with modern statistical design and analysis, to utilization of semantic techniques to accumulate and disperse the chemical information efficiently. The way these investigations inform digital curation and dissemination are highlighted in the following sections. We consider the generation of data from instrumentation in section 2 and the Crystallography supplies the major example used here to demonstrate different approach to data dissemination discussed in section 6. The need to capture and link metadata in this and other chemical laboratories is covered in section 3 and examples provided in sections 4 and 5. Wider community interaction is considered in section 7 and conclusions drawn in section 8.

2 Instruments on the Grid and the National Crystallography Service (NCS)

In the Grid-enabling of research in structural chemistry, we focused on chemical crystallography and relating this to the chemical and physico-chemical properties of the target materials, adopting a "high-throughput philosophy", processing large families of compounds while embedding the protocols for capture of metadata, date-stamping and adoption of agreed standards for archival and dissemination.

The CombeChem philosophy was applied to the operation of the EPSRC Chemistry Programme funded National Crystallography Service (NCS). This facility is a global centre of excellence with a long established service providing experimental structural chemistry resources for the UK chemistry community. The NCS involvement has provided an exemplar of how e-Science methodologies enhance user interaction with an operational service that provides resources to a distributed and varied user base and resulted in a set of recommendations for the construction of such an infrastructure (now being adopted by other EPSRC services). The NCS Grid facility has been deployed as a set of unified services that provide application, submission, and certificated secure, sample status monitoring, data access and data dissemination facilities for authenticated users, and is designed so that components currently under various stages of development from the project may be easily incorporated. Useful lessons have been learned in implementing security features for both the NCS systems, where a balance has been found between cost and a very secure network. In addition, remote and automated data collection procedures, based on a combination of robotics and scripted software routines, and coupled with automated structure solution and refinement, are presented.

2.1 Workflow Analysis

A first step towards designing such a complex system was the identification of the sequence of individual processes taken by users, service operators and samples, from an initial application to use the service to the final dissemination and further use of a crystal structure. All major activities, interactions and dependencies between the parties involved (both human and software components), may then be described as a workflow, from which an architecture that would accommodate all the processes could be designed.

The workflow for a typical service crystallography experiment is quite complex when considered at this level of granularity. For this reason it will not be reproduced here, -a thorough discussion is provided in reference [6]. A typical Grid, or web, service would only involve computing components (e.g. calculations, data retrieval services), hence the workflow involving these services is fairly trivial to derive and can be automated by an appropriate workflow engine. However, the service crystallography workflow also includes many manual operations, e.g. sending a sample to the service or mounting a sample on a diffractometer. From the analysis it was evident that the NCS Grid Service, in common with the few other scientific instruments on the Grid[7,8,9,10], is server-driven as opposed to purely computational Grid services that are generally orchestrated by the user.

The workflow gives rise to the design of a database which is core to the system and is capable of tracking a sample through the workflow. A sample is automatically given a different status in this database, according to its position in the workflow and each status has different authorisation conditions. The interplay between a generalised form of the workflow and the status of a sample is shown in Figure 1.

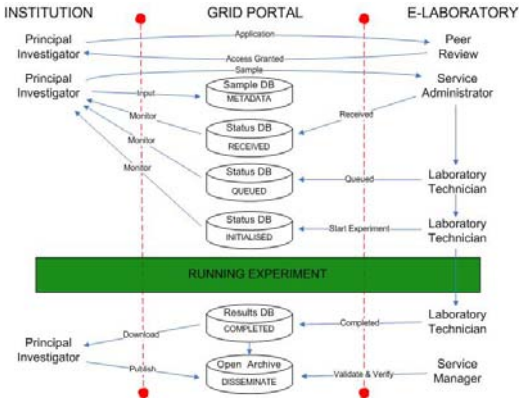


Figure 1

The X-ray diffractometer is normally controlled by bespoke software manually driven by the service operator via a Graphical User Interface (GUI). However, it is also possible to drive the diffractometer using command line calls, via an Advanced Program Interface (API). So, for the NCS Grid Service, scripts have been developed to drive the workflow normally carried out manually, which is essential as the experiment must be run automatically. As the experiment progresses raw data are deposited into a unique working directory on the NCS subnet system, to which the user has no direct access. The necessary experimental data are made available to the user by copying to a secure location on the DMZ server. The control script also makes calls to the sample/status database, at various key points during the experiment, to change the status of the sample being analysed.

2.2 Security and registration procedure

The NCS Grid service security infrastructure is designed in accordance with a Public Key Infrastructure[11] (PKI) policy. This requires the validity of each party involved in a transaction to be verified and authenticated using a system of X.509 digital certificates issued by a Certification Authority (CA) and a Registration Authority (RA). The issuing of certificates conforming to the X.509 specification requires adherence to a strictly-defined procedure[12]. Initially this was adopted, but credibility with the users required a slightly different approach. An alternative approach was devised[13] to avoid the requirement of users to install and use the relatively complex software used for the sign up and key management processes. The modified approach retains the software mechanisms, but handles the key generation centrally at the NCS. This deviates from a strict PKI in that user key-pairs as well as certificates are centrally generated, (i.e. by the NCS CA/RA), signed, and then securely transferred to the user, rather than relying on the user to perform the Certificate Signing Request (CSR) generation. the identity of the requestor and also to transmit the signed certificate and its corresponding passcode. As user generation of private keys becomes more commonplace and the supporting software more user friendly, the NCS intends to adopt standard CA/RA CSR practice. Users are required to re-register annually to obtain an allocation and new certificates are issued accordingly. It is therefore possible to update the security infrastructure at the same time, should it be considered necessary to update or integrate with other schemes.

2.3 User interaction during the experiment

The core of the NCS Grid service is the sample status database, which contains information on the position of the sample in the experimental workflow that may be updated by the system as processes are completed. A Status service written in PHP and running on the server visible to users, determines the DN of a user requesting access from their certificate and uses this to query the sample status database to obtain only the sample data owned by that DN.

At the point when the experiment is ready to start the service operator starts the experiment script, which automatically updates the status to *running* and provides a hyperlink in the status service that enables the user to participate with the experiment through the control service. The service operator may now leave the experiment for the user to monitor and/or guide and download the acquired data. On completion, the service personnel are responsible for the transfer of the complete archived dataset (raw and derived data) to an archival service for long term storage and retrieval if necessary. The NCS grid service uses the Atlas Datastore[14], based at the Rutherford Appleton Laboratory, UK. Approximately 1 Gb of data per day is transferred to the Atlas Datastore, who store the data with an off-site fire safe backup and migrate it on to new media as their service develops. Currently the data transfer is via FTP, but other front-ends to the Atlas Datastore are also provided, e.g. SRB[15].

3 Linking Experimental Data, Statistical Analysis and Publication@Source

We have used this combination of experiment and simulation as an exemplar of the advantages offered by linking publication to the original data. In a traditional journal publication, only the processed results are available, and there is no opportunity for the reader to examine their reliability or provenance. Indeed, there is usually not even the opportunity to see the exact numerical values plotted in a particular graph. A publication illustrating the linking of the original data to the published paper in the context of our Second Harmonic Generation experiments is in press; using the web version of this paper, it is possible to see and repeat all operations performed on the collected data to yield the final reported results.

4 Metadata, RDF, Smart Store and the Semantic Chemical Grid

Pervasive computing devices are used to capture live metadata as they are created in the laboratory, relieving the chemist of the burden of metadata creation. These data then feed into the scientific data processing. All usage of the data through the chain of processing is effectively an annotation upon it. By making sure everything is linked up through shared URIs, or assertion of equivalence and other relationships between URIs, scientists wishing to use these experimental results in the future can link back to the source (i.e., the provenance is explicit). CombeChem inherited traditional relational database technology to store experimental data and discovered the limitations in an environment where the user requirements are ever and rapidly changing.

As a Semantic Grid project, CombeChem has successfully deployed the latest Semantic Web technologies, including the RDF triplestore developed by the Advanced Knowledge Technology Interdisciplinary Research Collaboration (AKT IRC), to address the integration of diverse datastores. This was and continues to be a testing deployment, directly reflecting real world requirements, as different stakeholders own the different stores. There are significant gains to be had by this approach (reflecting Hendler's maxim: "a little semantics goes a long way"). While other groups are adopting XML formats (such as CML) for data interchange, CombeChem has taken advantage of these ideas but has moved to a higher level through widespread adoption of RDF. Knowledge technologies were applied through collaboration between Collaborative Advanced Knowledge Technologies in the Grid (CoAKTinG) and CombeChem, where the tools enhance the collaboration environment for chemists and help provide a complete digital record of the scientific process: the collaborative Semantic Grid.

5 SmartTea, Electronic Laboratory Notebooks and User-Centred Design

The pervasive computing aspects have focused on collecting data "at source", either from the handheld devices or from sensors recording experimental conditions. The former has been tackled through studying chemists at work in a laboratory and then designing a new device to support their work. Throughout the project, effort has been put into usability and representation of results for use by humans (HCI) as well as by machines; the SmartTea system and e-Bank are examples.

ELNs are currently attracting much interest. In a daylong symposium at an ACS National Meeting in 2004, the presentations from Southampton stood out from all the others in terms of scientific content and system design. "Smart tea" has been highly cited in the ELN and user design literature as an example of user-centred design by analogy. The requirements study was carried out in such a way that the designers and the users could communicate effectively and produce a truly user-friendly and effective system. The design approach, also used in the statistics teaching packages, demonstrated the need for context-sensitive systems. The monitoring of the laboratory environment has been made much more flexible and responsive to change by the adoption of the publication/subscribe model facilitated by the use of IBM micro-broker and MQTT middleware, for which we obtained significant publicity (including the BBC) for the use of mobile phones to monitor the laboratory.

The "back-end" for these laboratory systems, recording the processes undertaken, uses the same RDF technology as in the recording of information about the molecular species highlighted above. This shows the way to integrate the process information captured by the pervasive technology with the knowledge base about the materials, all using the Semantic Grid approach. The further analogy between the publication/subscribe approach and the publication@source provenance model

suggested new models for the dissemination of data as well as ideas, building on the e-Print OAI approach pioneered in Southampton, which CombeChem is taking forward (e.g., e-Crystals), together with the JSIC-funded e-Bank I, II and Repository for the Laboratory (R4L) projects.

6 OAI Repositories for data capture, management and dissemination

An end-to-end process demands a publication procedure at one end and the CombeChem project has built on this with the e-Bank project (http://www.ukoln.ac.uk/projects/ebank-uk) to produce the e-Crystals archive. This crystallographic e-Prints process obviates the need to pack scientific papers with vast amounts of data. The papers can concentrate on the presentation and discussion of ideas and the data is consulted only when required. An Open Archive has been developed to disseminate all the data accumulated during the course of the crystallographic experiment. This archive not only allows free and unhindered access to the data underpinning a scientific study but also publicises its content (including existence of the data) through established digital library protocols (e.g., OAI). The dissemination of results, held at the NCS service, that would otherwise remain hidden, will benefit structural researchers worldwide. However, not all data is "good data" (from a reputable source), so the provision of provenance tracking, as in e-Crystals, is essential for potentially un-refereed data.

6.1 The OAI crystal structure archive

The archive is a highly structured database that adheres to a metadata schema which describes the key elements of a crystallographic dataset. Current details of this schema can be found at http://www.ukoln.ac.uk/projects/ebank-uk/schemas/. The schema requires information on bibliographic and chemical aspects of the dataset, such as chemical name, authors, affiliation etc, which must be associated with the dataset for validation and searching procedures. As standards must be adopted in order for the metadata in the archive to be compatible with that already accepted and available in the public domain a tool for aiding the deposition process has been built. This tool performs the necessary file format transformations and operations necessary for presentation of the dataset to the archive. The elements of the schema and a brief description of their purpose are given in reference 6.

On completion of the crystal structure determination all the files generated during the process are assembled and deposited in the archive. For conventional publication purposes a crystal structure determination would normally terminate at the creation of a Crystallographic Information File (CIF)[16] and this file would be all that is required for submission to a journal, however this archive makes available ALL the underlying data. The metadata to be associated with the dataset is generated at this point, either by manual entry through a deposition interface or by internal scripting routines in the archive software which extract information from the data files themselves. All the metadata are then automatically assembled into a structured report and an interactive rendering of a Chemical Markup Language[17] (CML) file added for visualisation purposes (Figure 2).

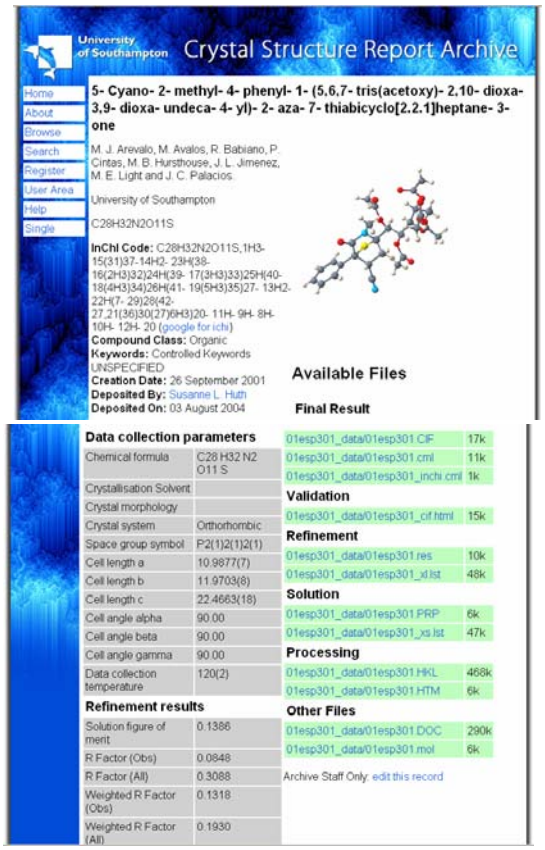


Figure 2

6.2 Metadata harvesting and value added services

When an archive entry is made public the metadata are presented to an interface with the internet in accordance with the Open Archive Initiative – Protocol for Metadata Harvesting (OAI-PMH)[18]. OAI-PMH is an accepted standard in the digital libraries community for the publication of metadata by institutional repositories which enables the harvesting of this metadata. Institutional repositories and archives that expose their metadata for harvesting using the OAI-PMH provide baseline interoperability for metadata exchange and access to data, thus supporting the development of service providers that can add value. Although the provision of added value by service providers is not currently well developed a number of experimental services are being explored. The eBank UK project has developed a pilot aggregator service[19] that harvests metadata from the archive and from the literature and makes links between the two. The service is built on DC protocols and is therefore immediately capable of linking at the bibliographic level, but for linking on the chemical dataset level a different approach is required. The Dublin Core Metadata Initiative (DCMI) provides recommendations for including terms from vocabularies in the encoded XML, suggesting: *Encoding schemes should be implemented using the xsi:type attribute of the XML element for property*. As there are not as yet any designated names for chemistry vocabularies, for the purposes of building a working example the project defined some eBank terms as designators of the type of vocabulary being used to describe the molecule. Thus a chemical formula would be expressed in the metadata record as:

<dc:subject
xsi:type="ebankterms:ChemicalFormula">C27 H48</dc:subject>

There are currently no journals publishing crystal structure reports that disseminate their content through OAI protocols. In order to provide proof of concept for the linking process the RSS feed for the International Union of Crystallography's publications website was used to provide metadata and crystal structure reports published in these journals were then deposited in the archive, thus providing the aggregator service with two sources of information. Aggregation is performed on the following metadata; author, chemical name, chemical formula, compound class, keywords and publication date, thus providing a search and retrieval capability at a number of different chemical and bibliographic levels. The demonstrator system, along with searching guidelines may be viewed and used at http://eprints-uk.rdn.ac.uk/ebank-demo/.

6.3 Data capture and management

Building on the advances made by the eBank project we are now developing repositories to drive efficient and complete capture of data as it is generated in the laboratory. Embedding the archive deposition process into the workflow in the laboratory in a seamless and often automated manner allows the acquisition of all the necessary files and associated metadata. This procedure is determined and driven by the experimental workflow and publication process and the 'Repository for the Laboratory' project (R4L: http://r4l.eprints.org) is in the process of workflow and publisher requirements capture. The repository driven experiment has the advantage that very accurate metadata can be recorded and a registration process has been designed whereby an experimenter can unambiguously assert that he/she performed a particular action at a particular time. A schematic of this project is shown in figure 3.

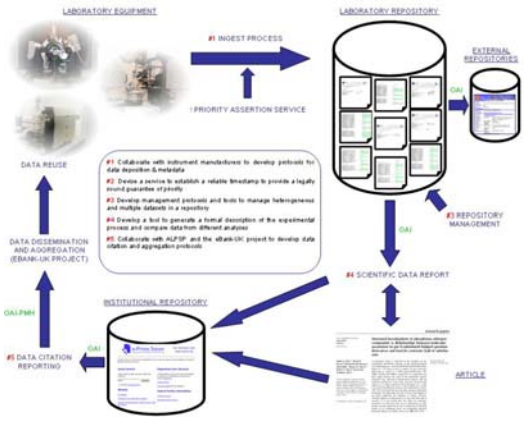


Figure 3

7 Outreach: CombeChem and e-Malaria

It is generally acknowledged that the public reputation of science is poor. In schools, science teaching can be uninspiring, science is perceived as boring, hard and irrelevant to people's lives, and the decline in the number of pupils choosing science courses is worrying for the science community and society at large. To address this difficulty, we have, as part of a project jointly supported by CombeChem and

JISC, developed an educational tool targeted at drug design for malaria. An integrated software environment combining web design, database development, and distributed computing has been developed. The software is aimed at A-level students of chemistry; the students are asked to design, using a sketch pad, chemical compounds that they can then submit for docking against a known malaria target. The score from the docked structure may then be used, together with molecular graphics, to refine further a potential drug. This software teaches the elements of molecular structure and intermolecular forces, with the added driver of targeting a serious illness.

Further development of this project through the South Eastern Science Learning Centre based at Southampton is planned. Diseases such as malaria, while being unprofitable to most pharmaceutical companies, make good choices for academic outreach projects. At a system level, software to be used by school students has to be designed differently from that used by university researchers and it must also be robust. The project has taken these factors on board and presents valuable lessons in how to achieve the secure integration of industrial strength programs into a "free" outreach environment.

8 The Pervasive Semantic Grid

Our deployment and use of pervasive computing technologies was informed by working alongside the infrastructure team of the Equator IRC, and this research is directly in line with the Next Generation Grids strategic report from the European Commission, with its attention to ambient intelligence. We have developed a particularly powerful combination of Grid and pervasive computing in that we do not just have the Grid meeting the physical world through the pervasive devices, but rather we have the physical world intersecting with the Semantic Grid: the experimental processes are themselves described as RDF graphs; the devices capturing experimental conditions do so in the form of semantic annotation; and interaction with the information is seen as annotation, that is, as enrichment through use. Hence the CombeChem vision provides a case study in the "Pervasive Semantic Grid". Much has been said about the Semantic Web but we will not see the benefits of Semantic Web technology unless programmes such as CombeChem actually use it.

The CombeChem team has achieved a real Semantic Web that enhances the ability to disseminate and curate data with appropriate context. A danger in the Grid community is that it is building within its own clique but the Grid is no good unless other people can join in. CombeChem is the real Grid because it has real users. We are changing the way in which science is done and ensuring that in the future our data and information will be used and useable!!

9 Acknowledgements

We acknowledge the tremendous support from the whole of the CombeChem and e-Bank teams, from the NCS, and colleagues in the Schools of Chemistry and Electronics & Computer Science. CombeChem was funded under the UK e-Science programme EPSRC grants GR/R67729/01 and EP/C008863/1. eBank is funded by JISC under the Semantic Grid and Autonomic Computing Programme.

10 References

[1] T. Hey, A. Trefethen, Cyberinfrastructure for e-Science, Science 308 (2005) 817–821.

[2] http://www.combechem.org

[3] R. Gledhill, S. Kent, B. Hudson, W.G. Richards, J.W. Essex, J.G. Frey, A Computer-Aided Drug Discovery System for Chemistry Teaching, J. Chem. Inf. Model, 2006, ASAP Article, DOI: 10.1021/ci050383q

[4] www.iupac.org/inchi

[5] D. De Roure, N. Jennings, N.R. Shadbolt, Research Agenda for the Semantic Grid: A Future e-Science Infrastructure; Technical Report UKeS-2002-02; National e-Science Centre: Edinburgh, UK, 2001; De Roure, N.R. Jennings, N.R. Shadbolt, The Semantic Grid: Past, Present, and Future, Proc. IEEE 93 (2005) 669–681.

[6] S.J. Coles, J.G. Frey, M.B. Hursthouse, M.E. Light, A.J. Milsted, L.A. Carr, D. De Roure, C.J. Gutteridge, H.R. Mills, K.E. Meacham, M. Surridge, E. Lyon, R. Heery, M. Duke and M. Day, M. (2006). An E-Science Environment for Service Crystallography from Submission to

Dissemination. *Journal of Chemical Information and Modeling* (doi:10.1021/ci050362w)

[7] R. Bramley, K. Chiu, J.C. Huffman, K. Huffman and D.F. McMullen, Instruments and Sensors as Network Services: Making Instruments First Class Members of the Grid, *Indiana University CS Department Technical Report 588*, 2003.

[8] http://nmr-rmn.nrc-cnrc.gc.ca/spectrogrid_e.html

[9] http://www.itg.uiuc.edu/technology/remote_microscopy/

[10] http://www.terena.nl/library/tnc2004-proceedings/papers/meyer.pdf

[11] A. Nash, W. Duane, C. Joseph, O. Brink, B. Duane, PKI: implementing and managing E-security, 2001, New York: Osborne/McGraw-Hill.

[12] R. Guida, R. Stahl, T. Blunt, G. Secrest, J. Moorcones, Deploying and using public key technology, *IEEE Security and Privacy*, 2004**,** *4*, 67-71.

[13] A. Bingham, S. Coles, M. Light, M. Hursthouse, S. Peppe, J. Frey, M. Surridge, K. Meacham, S. Taylor, H. Mills, E. Zaluska, Security experiences in Grid-enabling an existing national service, Conference submission to: eScience 2005, Melbourne, Australia.

([14]) http://www.e-science.clrc.ac.uk/web/services/datastore

([15]) http://www.sdsc.edu/srb/

([16]) I.D. Brown, B. McMahon, CIF: the computer language of crystallography**.,** *Acta Cryst*., 2002, *B58*, 317-324.

([17]) P. Murray-Rust, H.S. Rzepa, M. Wright, Development of Chemical Markup Language (CML) as a System for Handling Complex Chemical Content, *New J. Chem.*, 2001, 618-634.

([18]) C. Lagoze, H. Van de Sompel, M. Nelson, S. Warner, The Open Archives Initiative Protocol for Metadata Harvesting, Version 2.0. 2002, http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm

([19]) M. Duke, M. Day, R. Heery, L.A. Carr, S.J. Coles, Enhancing access to research data: the challenge of crystallography. Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries, 2005, 46 – 55, ISBN:1-58113-876-8

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Long-term Digital Metadata Curation

Arif Shaon

The Centre for Advanced Computing and Emerging Technologies (ACET)

The University of Reading, Reading, UK

Abstract

The rapid increase in data volume and data availability along with the need for continual quality assured searching and indexing information of such data requires efficient and effective metadata management strategies. From this perspective, the necessity for adequate, well-managed and high quality Metadata is becoming increasingly essential for successful long-term high quality data preservation. Metadata's assistance in reconstruction or accessibility of preserved data, however, bears the same predicament as that of the efficient use of digital information over time: long-term metadata quality and integrity assurance notwithstanding the rapid evolvements of metadata formats and related technology. Therefore, in order to ascertain the overall quality and integrity of metadata over a sustained period of time, thereby assisting in successful long-term digital preservation, effective long-term metadata curation is indispensable. This paper presents an approach to long-term metadata curation, which involves a provisional specification of the core requirements of long-term metadata curation. In addition, the paper introduces "Metadata Curation Record", expressed in the form of an XML Schema, which essentially captures additional statements about both data objects and associated metadata to aid long-term digital curation. The paper also presents a metadata schema mapping/migration tool, which may be regarded as a fundamental stride towards an efficient and curation-aware metadata migration strategy.

1. Introduction

Owing to an exponential increase in computing power and communication bandwidth, the past decade has witnessed a spectacular growth in volume of generated scientific data. Major contributors to this phenomenal data deluge include increasingly new avenues of research and experiments facilitated by the virtue of e-Science, which enables increasingly global inter-disciplinary collaborations of people and of shared resources needed to solve the new problems of science, engineering and humanities. In fact, e-Science data generated thus far from different areas, such as sensors, satellites, high-performance computer simulations and so on has been measured in the excess of terabytes every year and is expected to inflate significantly over the next decade. Several Terabytes of scientific data, generated from different scientific research and experiments conducted over the past 20 years and hosted by the Atlas Datastore of CCLRC's e-Science centre[1] provide an ideal of example of this data deluge.

This increasingly large volume of scientific data needs to be maintained (i.e. preserved) and highly available (i.e. published) over substantially long-periods of time in order to serve it to the future generations of scientists and researchers. This will, amongst other things, assist in avoiding the high cost of replicating data that will be expensive to regenerate and thereby aiding related experiments and researches in the foreseeable and distant future. Understandably, failure in ensuring continued access to good quality data, could potentially lead to its under-utilisation to a considerable extent. Conversely, efficiently managed and preserved information ascertains its proper discovery and re-use. This effectively bears the desirable potential for assisting in high quality future research and experiments in both same and cross-discipline environments, as well as other productive uses.

Evidently, one of the major challenges towards achieving efficient and continued use of valuable data resources is to ensure that their quality and integrity remains intact over time. This is driven by the changes in technologies and increased flexibility in their use that result in transforming and putting the integrity of very data they create at jeopardy. Therefore, with rapid evolution and enhancements in related technologies and data formats, the task of ensuring data quality for long periods of time, i.e.

[1] E-Science Centre, CCLRC - http://www.e-science.clrc.ac.uk/web

successful long-term[2] data preservation, may seem incredibly challenging.

Under the challenges set by the task of successful long-term data preservation, the word 'Metadata' is becoming increasingly prevalent, with a growing awareness of the role that it can play in accomplishing such a task. In fact, the digital preservation community has already envisaged the need of good quality and well-managed metadata for reducing the likelihood of important data becoming un-useable over substantially long periods of time (Calanag, Sugimoto, & Tabata, 2001). Furthermore, it has been recognized that metadata can be used to record information required to reconstruct or at the very least understand the reconstruction process of digital resources on future technological platform (Day, 1999).

Metadata's assistance in reconstruction or accessibility of preserved data, however, bears the same predicament as that of the efficient use of digital information over time: long-term metadata quality and integrity assurance notwithstanding the rapid evolvements of metadata formats and related technology. The only solution to this problem is employment of a well conceived, efficient as well as scalable long-term curation plan or strategy for metadata. In effect, curation has the ability to inhibit metadata from becoming out of step with the original data or undergoing additions, deletions or transformations, which change the meaning without being valid. In other words, in order to ascertain the overall quality and integrity of metadata over a sustained period of time, thereby assisting in ensuring continued access to high quality data (particularly within the e-Science context), effective long-term metadata curation is indispensable. With the evident necessity of long-term preservation of scientific data, the task of long-term curation of the metadata associated with that data is therefore equally important and beneficial in the context of e-Science.

This paper endeavors to provide a concise discussion of the main requirements of long-term metadata curation. In addition, the paper introduces "Metadata Curation Record", expressed in the form of an XML Schema, which essentially captures additional statements about both data objects and associated metadata to aid long-term digital curation. The paper also presents a metadata schema mapping/migration tool, which may be regarded as a fundamental stride towards an efficient and curation-aware metadata migration strategy.

2. Motivation

Over the past few years, several organized and arguably successful endeavors (e.g. The NEDLIB project[3]) have been made in order to find an effective solution for successful long-term data preservation. However, the territory of long-term metadata curation, although increasingly acknowledged, thus far, is even somewhat unexplored, let alone conquered. In fact, in most digital preservation or curation motivated workgroups and projects, necessity of long-term metadata curation is seen relegated to the backseat and deemed secondary, mainly due to lack of awareness of the criticality of the problem. As a result, no acceptable methods exist to date for effective management and preservation of metadata for long periods of time. The Digital Curation Centre (DCC), UK[4] in conjunction with the e-Science Centre of the CCLRC however, aims to solve this rather cultural issue. The work presented in this paper contributes to the long-term Metadata Curation activity of the DCC.

3. Metadata Defined

The word "metadata" was invented on the model of *meta-philosophy* (the philosophy of philosophy), *meta-language* (a language to talk about language) etc. in which the prefix *meta* expresses reflexive application of a concept to itself (Kent and Schuerhoff, 1997). Therefore, at the most basic, metadata can be considered as data about data. However, as conformity of the middle term ("about") of this definition is crucial to a common understanding of metadata, this classical and simple definition of metadata has become ubiquitous and is understood in different ways by many different professional communities (Gorman, 2004). For example, from the bibliographic control outlook, the focus of "aboutness" is on the classification of the source data for identifying the location of information objects and facilitating the collocation of subject content. Conversely, from the computer science oriented data management perspective, "aboutness" may well emphasize on the enhancement of use in relation to the source data. Moreover, this metadata or "aboutness" is synonymous with its context in the sense of contextual information.

[2] The time period over which continued access to digital resources with accepted level of quality despite the impacts of technological changes is beneficial to both the user base and curatorial organization(s) of the resources.

[3] Networked European Deposite Library - http://www.kb.nl/coop/nedlib/

[4] Digital Curation Centre, UK – http://www.dcc.ac.uk

Nevertheless, in light of its acknowledged role in the organisation of and access to networked information and importance in long-term digital preservation, metadata may be defined as structured, standardized information that is crafted specifically to describe a digital resource, in order to aid in the intelligent, efficient and enhanced discovery, retrieval, use and preservation of that resource over time. In the context of digital preservation, information about the technical processes associated with preservation is an ideal example of metadata.

4. Metadata Curation

In effect, the phrase "Metadata Curation" is an integral part of the phrase "Digital or Data Curation", which has different interpretations within different information domains. From the museum perspective, data curation covers three core concepts – data conservation, data preservation and data access. Access to data or digital information in this sense may imply preserving data and making sure that the people to whom the data is relevant can locate it - that access is possible and useful. Another interpretation of the phrase "Data Curation" may be the active management of information, involving planning, where re-use of the data is the core issue (Macdonald and Lord, 2002).

Therefore, in essence, long-term data or digital curation is the continuous activity of managing, improving and enhancing the use of data or other digital materials over their life-cycle and over time for current and future generations of users, in order to ensure that its suitability sustains for its intended purpose or a range of purposes, and it is available for discovery and re-use.

In light of the above construal of digital preservation, Metadata curation may be defined as an inherent part of a digital curation process for the continuous management of metadata (which involves its creation and/or capturing as well as assuring its overall integrity) over the life-cycle of the digital materials that it describes in order to ascertain its suitability for facilitating the intelligent, efficient and enhanced discovery, retrieval, use and preservation of those digital materials over time.

5. Requirements of Metadata Curation

The efficacy of Metadata curation largely relies upon successful implementation of a number of requirements. Although metadata curation requirements may be quite different according to the type of data described, the information outlined below attempts to provide a general overview of the main requirements.

5.1 Metadata Standards

The Digital Preservation professionals have already perceived the necessity of metadata formats/standards[5] in forestalling obsolescence of metadata (hence obsolescence of the actual data or resource), due to dynamic technological changes. In the context of long-term data curation, it is essential that the structure, semantics and syntax of Metadata conform to a widely supported standard(s), so that it is effective for the widest possible constituency, maximises its longevity and facilitates automated processing.

As it would be impractical to even attempt to determine unequivocally what will be essential in order to curate metadata in the future, the metadata elements should reflect (along with other relevant information such as, metadata creator, creation date, version etc.) necessary assumptions about the future requirements in that regard. Furthermore, the metadata elements should be interchangeable with the elements of other approved standards across other systems with minimal manipulation in order to ensure metadata interoperability. This will consequently aid in minimization of overall metadata creation and maintenance cost. It may also be advantageous to define specific metadata elements that portray metadata quality.

5.2 Metadata Preservation

Metadata curation requires metadata to be preserved along with data in order to ensure its accurate descriptions over time. To date, the dominant approach to long-term digital preservation has been that of migration. Unfortunately, it does pose the notable danger of data loss or in some cases the loss of original appearance and structure (i.e. 'look and feel') of data as well as being highly labour intensive. However, in the context of metadata preservation, 'look and feel' of metadata (e.g. differing date/time formats) is not as imperative as that of the original data as long as it maintains its aptness for describing the original data accurately over time. Therefore, albeit the existence and availability of Emulation (which seeks to solve the problem of data 'look and feel' loss by mimicking the hardware/software of the original data analysis environment) Migration would appear to be a better solution for long-term Metadata preservation. Having said that, if a superior or

[5] Fundamentally, a metadata standard or specification is a set of specified metadata elements or attributes (mandatory or optional) based on rules and guidance provided by the governing body or organisation(s).

alternative preservation strategy is proposed, this would also be worth considering as both Emulation and Migration have received criticism for being costly, highly technical, and labour intensive.

However, a classic unresolved metadata migration issue is that of tracking and migrating changes to the metadata itself. This issue is likely to arise when currently used metadata standards/formats change and/or evolve in the future. For example, an element contained within a contemporary metadata format might be replaced in or even excluded from the newer versions of that format, thus incurring the problem of migrating the information under that element to its corresponding element(s) (if any) of the new format. Therefore, in order to successfully curate Metadata, a curation aware migration strategy needs to facilitate migration of (ideally from old formats to new formats) and tracking/checking changes (i.e. new formats to old formats) to metadata between different versions of its format but also be flexible for addition of further requirements.

5.3 Metadata Quality Assurance

As highlighted earlier in this paper, quality assurance of metadata is an integral part of long-term metadata curation. It needs to be ensured that appropriate quality assurance procedures or mechanisms are in place to eliminate any quality flaws in a metadata record and thereby, ascertain its suitability for its intended purpose(s). As identified in (JISC, 2003), incorrect and inconsistent content of metadata significantly lower the overall quality of metadata. This inaccuracy and inconsistency in metadata often occur due to lack of validation and sanity checking at the time of metadata entry as well as metadata creation guidelines respectively. Non-interoperable metadata formats and erroneous metadata creation and/or management tools are also considerable contributors to such metadata quality flaws.

In general, quality of a metadata record is measured by the degree of consistency with and/or accuracy in reference to the actual dataset and conformance to some agreed standard(s). Therefore, digital metadata curation process, at least the part or module contributing to metadata quality assurance should be executed on the following two levels:

- Semantic curation - organizing and managing the meaning of metadata, i.e. ensuring semantic validity of metadata to ensure meaningful description of dataset.

- Representational curation - organizing and managing the formal representation (and utilization) of metadata, i.e. ascertaining structural validity against metadata schema(s) and re-usability of metadata.

While representational curation (i.e. structural validation) typically begins during or after creation of metadata records, semantic curation can in fact start even before metadata records come into existence. The importance of semantic curation is easily deducible from the fact that in order to ensure effective assistance of metadata in long-term digital curation, metadata curation (i.e. semantic curation) should begin at the very outset of metadata lifecycle by developing or adopting a curation-aware metadata framework or ontology. It is also useful to have a rich set of functionality for metadata validation incorporated within metadata creation and management tools.

5.4 Metadata Versioning

Throughout the vibrant process of long-term metadata curation, metadata is prone to be volatile. This volatility may well be caused by updating of metadata that can involve amendments to or deletions from existing metadata records. However, previous versions of metadata may need to be retrieved in order to obtain vital information (e.g. in the case of annotation - who made the annotation and which version(s) of a value it applies to) about the associated preserved information if required. It is therefore essential to be able to discriminate between metadata in different states, which arise and co-exist over time by versioning metadata information.

5.5 Other Requirements

Aside from the requirements outlined above, long-term metadata curation need take the following additional issues into account:

- Metadata Policy: A set of broad, high-level principles that form the guiding framework within which the Metadata curation can operate, must be defined. The Metadata Policy would normally be a subsidiary policy of the organizational data policy statement and should reference the rules with reference to legal and other related issues regarding the use and preservation of data and metadata, as governed by the data policy statement.

- Audit Trail & Provenance Tracking: Metadata Curation Process should ensure recording of information with required granularity and facilitate necessary means to track any significant changes (e.g. provenance change) to both data and metadata over their life cycles.

This will, amongst other things, help provide assurance as to the reliability of the content information of both data and associated metadata.

- Access Constraints & Control: Appropriate security measures should be adopted to ensure that the metadata records have not been compromised by unauthorized sources, thereby ensure the overall consistency in the metadata records. Furthermore, verification of the metadata records' authenticity before they are ingested into the system for long-term curation is also a crucial requirement. Techniques such as digital signatures, checksum may be employed for this function. Fixity information, as defined within the OAIS framework (OAIS, 2002), is an ideal example that advocates the use of such techniques or mechanisms.

6. Metadata Curation Record

It would not be an overstatement to regard the term "information" as highly crucial as well as instrumental in the context of long-term digital curation. To elaborate, success of a long-term curation strategy predominantly relies on sufficient and accurate information about the resources being curated. Under the influence of this observation, the "Metadata Curation Record" (hereafter referred to as a MCR) has been constructed in the form of an XML Schema, which aims to record additional statements about both data objects and associated metadata to aid long-term digital curation.

In other words, the MCR essentially pursues two primary objectives. First objective is to capture as much information about a digital information object as possible in order to assist its long-term preservation, curation and accessibility. This objective may well echo the objectives of many widely used long-term preservation motivated XML schemas (e.g. PREMIS[6]). The second objective, on the other hand, is a feature that may not be discerned in most of contemporary metadata standards. That is to provide additional statements about the metadata record itself, thereby supporting long-term curation of that record. In a digital curation system, the metadata ingest interface and/or metadata extraction tools/services can be developed based on the MCR to ensure that appropriate and sufficient metadata is acquired to aid in the curation of both data and metadata.

The approach employed to construct the curation record involved examining a range of different existing well-known metadata schemas, such as Dublin Core[7], Directory Interchange Format (DIF)[8], DCC RI Label[9], CCLRC Scientific Metadata Model Version 2 (Sufi & Mathews, 2004) and IEEE Learning Object Metadata (LOM)[10] and importing the most relevant elements (in terms of curation, preservation and accessibility) from them. The rationale for this approach was to utilising existing resources and thereby avoiding wheel reinvention as much as possible.

6.1. Overview

In general, as depicted in figure 1, the elements contained within the Metadata Curation Record, are divided into four different categories: General, Availability, Preservation and Curation.

Firstly, the "General" category represents all generic information (e.g. Creator, Publisher, Keywords, etc.) about a data object. This category of elements is primarily required for presenting an overview of the digital object to its potential users. In addition, the elements (e.g. keywords, subject) that record keywords related information; may well be used to aid in keywords based searching for scientific data across disparate sources. Secondly, the "Availability" elements provide information with regards to accessing the data object, checking its integrity and any access or use constraints associated with it.

The "Preservation" category presents information that will assist in long-term preservation and accessibility of digital objects. Of particular mention is the OAIS compliant (OAIS, 2002) "Representation Information Label", which captures information required to enable access to the preserved digital object in a meaningful way. The use of RI label can be recursive, especially in cases where meaningful interpretation of one RI element requires further RI. This recursion continues until there is sufficient information available to render the digital object in a form the OAIS Designated Community can understand.

[6] PREservation Metadata: Implementation Strategies -http://www.oclc.org/research/projects/pmwg/

[7] Dublin Core Metadata Initiative - http://dublincore.org/

[8] DIF Writer's Guide - http://gcmd.gsfc.nasa.gov/User/difguide/difman.html

[9] DCC Info Label Report - http://dev.dcc.ac.uk/twiki/bin/view/Main/DCCInfoLabelReport

[10] IMS LOM - http://www.imsproject.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html

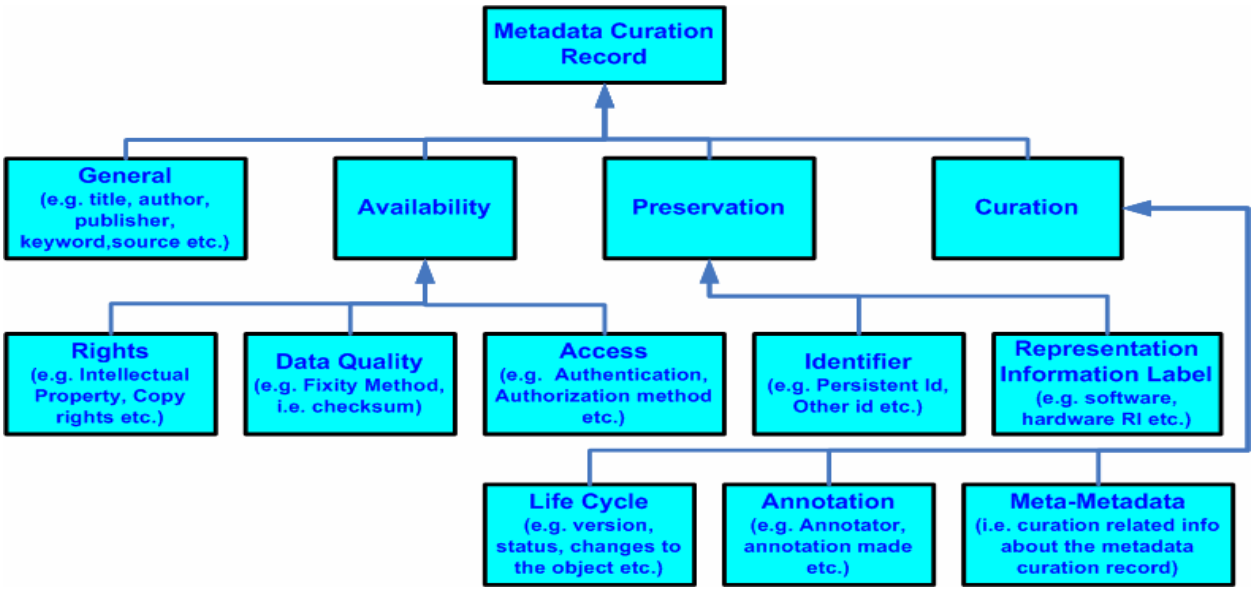


Figure 1: Abstract level view of the Metadata Curation Record

Finally, the "Curation" category elements provide information about life cycle and other related aspects (e.g. version, annotation) of the digital object and the metadata record itself, which may be used in efficient and long-term curation of both the digital object and its metadata. Of particular significance is the "LifeCycle" elements that are defined to record all changes along with information about who/what (e.g. factors, people etc.) conducted or was responsible for the changes that a digital object undergoes throughout its life cycle. This type of information is vital for implementing some crucial curation related functionalities, such as provenance tracking (as specified in the OAIS model) and audit trailing that are essential for checking and ensuring quality and integrity of data objects.

Furthermore, "MetaMetadata" element, in this category is dedicated towards capturing information required to efficiently curate the MCR itself. It has its own "General" (e.g. Creator, Indexing), "Preservation" (e.g. Identifier, Representation Information), "Availability" (e.g. Metadata Quality, Access, Rights etc.) and "Curation" (e.g. Event) category elements. In a Metadata Curation system, the "MetaMetadata" elements would be implemented as a separate schema complimenting the MCR consisting of the rest of the elements.

7. Metadata Schema Mapping Tool

Successful long-term metadata curation demands curation-aware migration strategy in order to cope with the metadata migration issue (see 5.2) that arises from metadata schema/format evolution. The metadata schema mapping tool that has been developed using JAVA technologies, aims to resolve this issue by effectively facilitating easy, semi-automatic migration of metadata between two co-existing versions of its format. The tool employs an efficient regular expression driven matching algorithm to determine all possible matches (direct or indirect) between two versions of a metadata schema irrespective of their types (i.e. XML or Relational), calculates mapping rules based on the matches and finally migrates metadata from the source schema to the destination schema.

7.1 Rationale

There are numerous commercial and non-commercial database schema mapping or migration tools available at present. Most of these tools enable users, to varying extent, to automatically find matches between two database and/or XML schemas and migrate and/or copy data across based on the matches. Examples of such tools include, Altova Map force[11], SwisSQL[12], etc. Many of these tools also facilitate interoperability between different databases by allowing users to perform cross-database schema migration, such as migration from Oracle to DB2, MS SQL to Oracle etc. Naturally, the existence of these tools may somewhat question the necessity and the motivation for the Metadata Schema Mapping Tool.

[11] Altova MapForce - http://www.altova.com/download_mapforce.html

[12] SwissSQL - www.swissql.com/

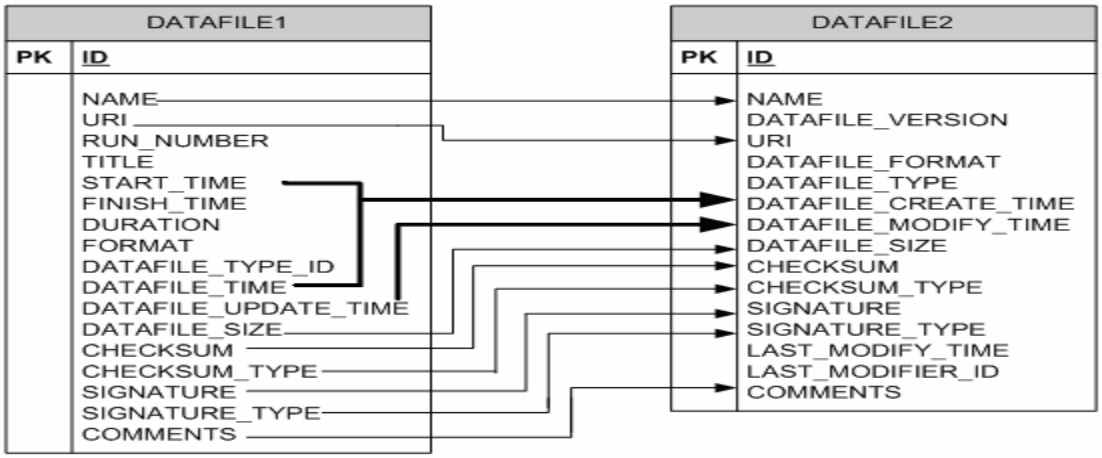


Figure 2: Direct & indirect matches between two versions of DATAFILE table

In response to this question, it would not be an overstatement to say that the inability of currently available tools to find indirect or non-obvious matches between two schemas essentially justifies the necessity and confirms the uniqueness of the Metadata Schema Mapping tool. To illustrate, two database tables of two versions of ICAT schema of CCLRC[13], as shown in figure 2, may be considered. Commercial tools will only be able to determine the direct matches (as indicated by the thin arrows from source table to destination table) between these two tables. Here, the term "Direct Match" refers to as exact duplicate or replica of a field or column in a database table in terms of the name and type of that particular field or column.

These two database tables together provide an ideal example of the aforementioned metadata migration issue, i.e. migrating changes to the metadata itself, especially in cases, where one or more crucial elements in the old format do not have any directly corresponding elements in the new format. For example, the field "DATAFILE_UPDATE_TIME" in the version 1 of DATAFILE table has been changed to "DATAFILE_MODIFY_TIME" in version 2 (indicated by a solid arrow in the diagram). Currently available tools will not be able to establish any link between these two fields as they only search for direct matches as shown in diagram 2.

However, looking up in any English Dictionary will help one easily determine that the word "UPDATE" in the former field is in fact synonymous to the word "MODIFY" in the latter, hence declare the aforementioned two fields as "in-direct match". The Metadata Schema Mapping tool is capable of doing exactly that along with other features, such as determining matches between two tables based on their relationship (i.e. foreign key based relationship) with other tables in corresponding schemas, in both cross-database and cross-schema format oriented settings.

In addition to its role in metadata curation, the mapping tool has the ability to play a perceivably significant role in any data management environment administering considerably large and evolving data sets. In such a dynamic environment, data evolution often results in evolution of underlying schema(s) as well as change of databases employed (e.g. MS Access to Oracle), to assist and maintain the changes to the datasets. The schema mapping tool provides an easier and relatively less labour-intensive means (than the commercial tools) of identifying and reconciling complex and "non-obvious" differences between schemas. Thereby, the tool effectively facilitates more accurate migration of data from an old version of a schema to its newer version irrespective of the schema type and/or the database they reside in, while enabling the use of both versions. This will effectively make the accessibility of the datasets to the users more declarative as they would be able to pose queries to the datasets based on a version of the schema they are familiar with. From this perspective, the use of the tool is indeed beneficial to the efficient management of ever-expanding scientific data generated from various e-Science activities.

[13] The ISIS ICAT is a metadata catalog of the previous 20 years of data collected at the ISIS facility. The database schema used by them is based heavily on the CCLRC Scientific Metadata Model version 2 (Sufi & Mathews, 2004).

8. Future Work

There is certainly a great deal of scope available for further advancement as well as innovation in terms of development of an efficient metadata curation strategy, concrete tool sets and so on. However, as the domain of long-term metadata curation has yet to be completely explored, it is difficult to unequivocally set a limit for the work to be done for a fully potent long-term metadata curation strategy. Nevertheless, key future activities of the project will include the development of a metadata curation model, which will effectively address the core requirements of long-term metadata curation. The model will essentially encompass a curation-aware metadata framework based on the MCR, efficient post-creation metadata quality assurance mechanisms and suitable metadata versioning techniques, amongst other things.

The first draft of the model has already been designed as an extension to the OAIS reference model and is currently being assessed for possible improvements. The OAIS defined archival information preservation functions are not, however, within the scope of the Metadata Curation Model, although the model has explicit reference to some of those functions. Furthermore, the model is only focused on the curation of metadata and does not assume the responsibility of curation of the data that the metadata describes.

9. Conclusions

Efficient and effective long-term metadata curation is a key component of successful preservation, enrichment and access of digital information in the long term. A preliminary research (Shaon, 2005) for this project revealed that majority of current metadata standards, systems and approaches (relevant to the context of metadata curation) in existence do not address the full set of metadata curation requirements as outlined in this paper. This profoundly addresses the necessity of curation-aware metadata standards, metadata management standards and system, which would effectively aid in developing a viable strategy for long-term metadata curation. Developing new standards for both the metadata and metadata management realm, however, would not be an efficient strategy. Therefore, a specification of extensions needed to aid metadata curation for existing standards and systems was recommended and seen as a fruitful area of both the work presented in this paper and future work of the project. The Metadata Curation Record and the Metadata Schema Mapping tool may therefore be seen as a union of best features of existing metadata standards and metadata mapping tools respectively. Nevertheless, the work should certainly be regarded as initial steps towards developing an efficient strategy for long-term metadata curation that would benefit any discipline concerned with long-term data preservation, such as e-Science.

References & Bibliography

Calanag, M.L., Sugimoto, S., & Tabata, K. (2001): A metadata approach to digital preservation. In: Proceedings of the International Conference on Dublin Core and Metadata Applications 2001 (pp. 143-150). Tokyo: National Institute of Informatics – http://www.nii.ac.jp/dc2001/proceedings/product/paper-24.pdf

Day, M. (1999): *Metadata for digital preservation*: *an update*, Ariadne Issue 22, 1999 - http://www.ariadne.ac.uk/issue22/metadata/intro.html

Gorman, G. E. (2004): International Yearbook of Library and Information Management, 2003-2004, metadata applications and management, facet publishing, 2004, part 1, pages 1-17.

JISC, (2003): *Quality Assurance For Metadata*, QA Focus Document, QA Focus, a JISC-funded advisory service supporting JISC 5/.99 projects 2003 -http://www.ukoln.ac.uk/qa-focus/documents/briefings/briefing-43/briefing-43-A5.doc

Kent, J. and Schuerhoff, M. (1997): *Some Thoughts About a Metadata Management System, 1997*, Statistics Netherlands – http://www.vldb.org/archive/vldb2000/presentations/jarke.pdf

Macdonald, A. and Lord, P. (2002): *Digital Data Curation Task Force Report of the Task Force Strategy Discussion Day*, November 2002 - http://www.jisc.ac.uk/uploaded_documents/CurationTaskForceFinal1

OAIS (2002): *Reference Model for an Open Archival Information System (OAIS)*, CCSDS Blue Book. Issue 1. January 2002 - http://public.ccsds.org/publications/archive/650x0b1.pdf

Shaon, A. (2005): Long-term Metadata Management & Quality Assurance in Digital Curation, MSc Dissertation, CCLRC ePublications Archive, 2005- http://epubs.cclrc.ac.uk/bitstream/897/MSc_Dissertation.pdf

Sufi, S. and Matthews, B. (2004): *The CLRC Scientific Metadata Model Version 2*, CCLRC ePublications Archive, August 2004, – http://epubs.cclrc.ac.uk/bitstream/485/csmdm.version-2.pdf

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Developing Lightweight Application Execution Mechanisms in Grids

Ligang He, Martin Dove, Mark Hayes, Peter Murray-Rust, Mark Calleja, Xiaoyu Yang, Victor Milman*
University of Cambridge, Cambridge, UK
**Accelrys Inc., Cambridge, UK*
lh340@cam.ac.uk

Abstract

CASTEP is an application which uses the density functional theory to calculate atomistic structure and physical properties of materials and molecules. This paper investigates the execution mechanisms for running CASTEP applications using grid resources. First, a lightweight execution environment is developed for CASTEP, so that the CASTEP applications can be run on any grid computing resource with the supported platform even without the CASTEP server being installed. Then, two execution mechanisms are developed and implemented in this paper. One is based on Condor's file transfer mechanism, and the other based on independent data servers. Both mechanisms are complete with the GridSAM job submission and monitoring service. Finally, the performance of these mechanisms is evaluated by deploying them on CamGrid, a university-wide condor-based grid system at the University of Cambridge. The experimental results suggest that the execution mechanism based on data servers is able to deliver higher performance; moreover, if multiple data replicas are provided, even higher performance can be achieved. Although these mechanisms are motivated by running CASTEP applications in grids, they also offer excellent opportunities to be applied to other e-science applications as long as the corresponding lightweight execution environments can be developed.

1. Introduction

Grid systems are becoming a popular platform for processing e-science applications [2][3]. CASTEP is a software package used for atomistic quantum-mechanical modeling of materials and molecules [7]. In order to run CASTEP applications, the CASTEP server is required to provide the necessary execution environment.

Currently, a popular approach to running CASTEP applications is to employ the Materials Studio Modeling from Accelrys Inc [12]. Materials Studio Modeling 4.0 is a Client/Server software environment [12]. At the client side, the users can create models of molecular structures and materials. The model information is transferred to the server side, where the calculations are performed and the results are returned to the users. The CASTEP server component is included in the server side. In the Grid environments, however, it cannot be assumed that Materials Studio is installed on all computational resources. This presents a challenge of harvesting the computational grid resources that have no pre-installed CASTEP server environments. This paper tackles this problem by developing a lightweight execution environment for CASTEP applications. The lightweight environment includes the CASTEP executables, shared libraries, relevant configuration files, license checking-out executables and license data, etc.

The execution mechanisms developed in this paper are implemented and deployed on CamGrid, a university-wide grid system at the University of Cambridge [1]. CamGrid is based on the Condor system [8], in which a federated service is provided to flock the condor pools located in different departments. The jobs submitted to a local Condor pool may be dispatched to another flocked remote pool depending on the resource availability [9].

GridSAM is a job submission and monitoring web service [6]. One of advantages of GridSAM is that the GridSAM client and server communicate with each other through the standard SOAP protocol. Moreover, GridSAM can be connected to different distributed resource managers such as Condor [4]. In this paper, GridSAM is deployed in the CamGrid and configured to distribute the jobs for execution through Condor. In the configuration GridSAM web service is available to the Internet, and the users can submit jobs over the Internet to the GridSAM service, where the jobs are further submitted by GridSAM to a condor job submission host. Then the job submission host further schedules the jobs to the remote exeute cmachines in CamGrid.

This paper investigates two approaches to transferring a lightweight CASTEP execution environment. One approach is to make use of the Condor's file transfer mechanism. In the Condor submission description file, it can be specified what files need to be transferred to the execute nodes. The job submission host will copy these files over once the execute nodes have been determined.

Another approach is to setup independent data servers for a lightweight CASTEP execution environment. When a job has been launched on execute nodes, it first stages in the necessary CASTEP execution environments and then starts running.

These CASTEP execution mechanisms are presented in detail in this paper. Their performance is evaluated on CamGrid in terms of various metrics, including makespan, average turnaround time, overhead and speedup. The performance is also compared against that achieved by the Materials Studio mechanism.

The work developed in this paper is motivated by the MaterialsGrid project being conducted at the University of Cambridge [13]. The MaterialsGrid project aims to create a dynamic database of materials properties based on quantum mechanical simulations run in grid environments. CASTEP is a typical example of a computational application involved in this project.

2. Lightweight Execution Environment

After installing Materials Studio Modeling 4.0 on a machine, the CASTEP execution environment is built. By tracing the calling path of the CASTEP executables, the necessary files and settings for running CASTEP applications are identified in this paper.

All files required to run a CASTEP application can be divided into the following three categories.

1. User-defined files: cell file and parameter file

CASTEP application needs to take as input the information about atomistic system users want to study. The information is summarised in two text files: *model*.cell and *model*.param (cell and param are the file extensions). The cell file defines the molecular system of interest, and the parameter file (.param) defines the type of calculation to be performed. These two files can be generated using Materials Studio Modeling 4.0 [12] or by the CML transformation toolkit [11][10].

2. Application-dependent system files: potential files

When calculating the property of a given model, the pseudopotential files corresponding to the elements composing the structure will be used. A pseudopotential file for an element describes the screened electrostatic interaction between valence electrons and ionic core for that element. The pseudopotential files for all elements in the periodic table are provided in the Materials Studio Modeling package.

3. Application-independent system files:

The files in this category are required by all calculations. These include:

- CASTEP executables and shell scripts – CASTEP executables are the programs for performing the actual calculations. These executables are wrapped in shell scripts, in which the environmental variables as well as other system configurations are defined and different CASTEP executables are invoked in different scenarios.
- Shared libraries – these libraries are required at run time by the CASTEP executables.
- License checking-out executables and relevant license data – the commercial version of Materials Studio Modeling 4.0 needs to check out the licence from the pre-installed license server before being able to run the CASTEP executables. These license-relevant files are included in a license pack, which is located in a directory separated from the Material Studio package. (Please note that a simpler license mechanism is applied to academic versions of CASTEP for the UK scientists; in this case the CASTEP executables do not need to check out the license files).

When installing the Materials Studio and the license Pack, the system configurations need be specified, such as the home directory of the installation, the location of the license server and license files. Under the Condor job manager, however, all files are placed in a single temporary working directory. Therefore, in order to successfully access the files in a execute node, these configurations need to be changed to adapt to the Condor working location.

After all necessary changes are made, the required files for running CASTEP applications are packed into tar files. The lightweight execution environment can be constructed in a execute node by unpacking these tar files. The total size of the Materials Studio and the License Pack package is 884 Megabytes. The size of the tar files for the lightweight CASTEP execution environment is reduced to 105 Megabytes.

3. Execution Mechanisms in Grids

In this section, different CASTEP execution mechanisms are presented in detail. The execution mechanisms differentiate from each other mainly by their approaches to dealing with the transferring of the CASTEP lightweight environment.

3.1 Execution Mechanism Based on Condor's File Transfer Mechanism

When connecting GridSAM with the Condor job manager, two major configuration files need to be edited. One is the jobmanager.xml file, which specifies the job manager and its settings used to distribute the jobs received by GridSAM. The other is the file classad.groovy, in which the values of the ClassAd attributes of a Condor job can be specified.

In the classad.groovy file, the transferred input files are specified, which include the user submitted cell file

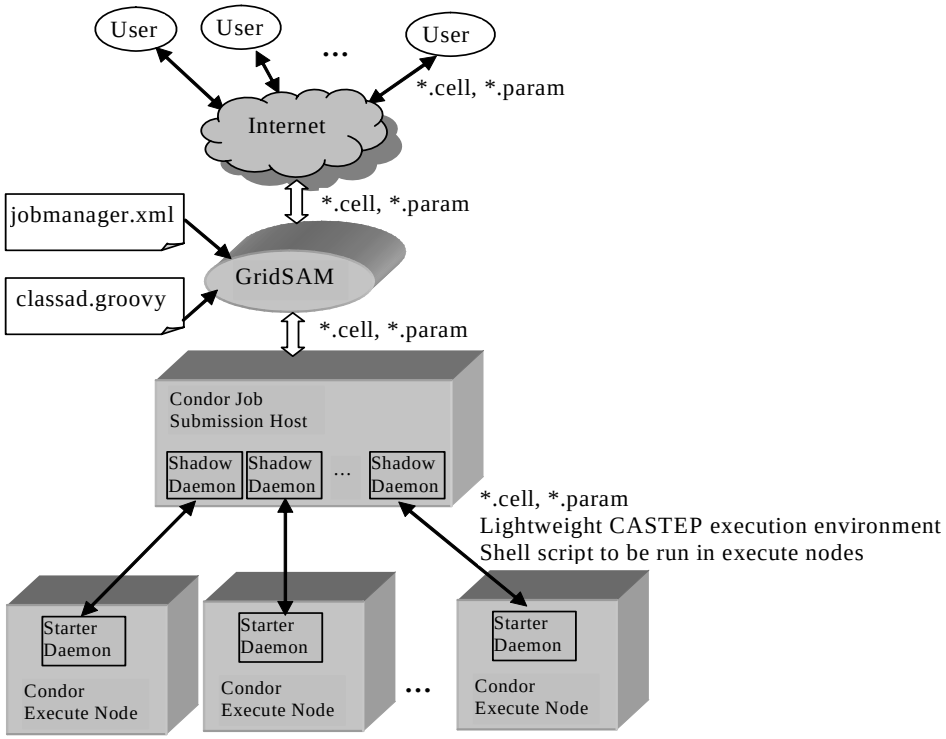


Figure 1. Application execution mechanism based on Condor's file transfer mechanism

```
1.  FILE1='MaterialsStudio.tar'
2.  FILE2='LicensePack.tar'
3.  FILE3='Al_00.usp'
4.  FILE4='O_00.usp'
5.  flag=`expr $RANDOM % 2`
6.  if [ $flag -lt 1 ]
7.  then
8.      HOST=$HOST1
9.      PORT=$PORT1
10. else
11.     HOST=$HOST2
12.     PORT=$PORT2
13. fi
14. ftp -n $HOST $PORT << END_SCRIPT
15.     quote USER $USER
16.     quote PASS $PASSWD
17.     get $FILE1
18.     get $FILE2
19.     get $FILE3
20.     get $FILE4
21.     quit
22. END_SCRIPT
23. tar -xf $FILE1
24. tar -xf $FILE2
25. Call the script RunCASTEP.sh, which is included in
    the MaterialsStudio.tar, to calculate the specified
    model, Al2O3
```

Figure 2. Execution flow of the data server-based execution mechanism in an execution node

and the parameter file as well as the tar files for lightweight CASTEP execution environment. This classad.groovy file is used by GridSAM to construct the job submission description file for Condor. When the constructed Condor submission description file is submitted in the job submission host (where the tar files for the lightweight CASTEP execution environment are also stored), the Sched daemon sends the job advert to matchmaker and then the matchmaker looks for the execute node whose resource advert can match the job advert. After the suitable node is found, the Sched daemon starts a Shadow daemon acting on behalf of the submitted job, while the Start daemon at the execute node spawns a starter daemon to represent the process which performs the actual calculation. After the communication between the Shadow daemon and Starter daemon has been established, the specified input files are transferred from the job submission host to the execute node. The steps of constructing the CASTEP environment (i.e., unpacking the transferred tar files) and performing the actual CASTEP calculations are wrapped in a shell script, which is the executable sent to the execute nodes.

This execution mechanism is illustrated in Fig.1.

3.2 Execution mechanism based on independent data servers

In this mechanism, the independent data servers are set up to accommodate the tar files for the lightweight CASTEP

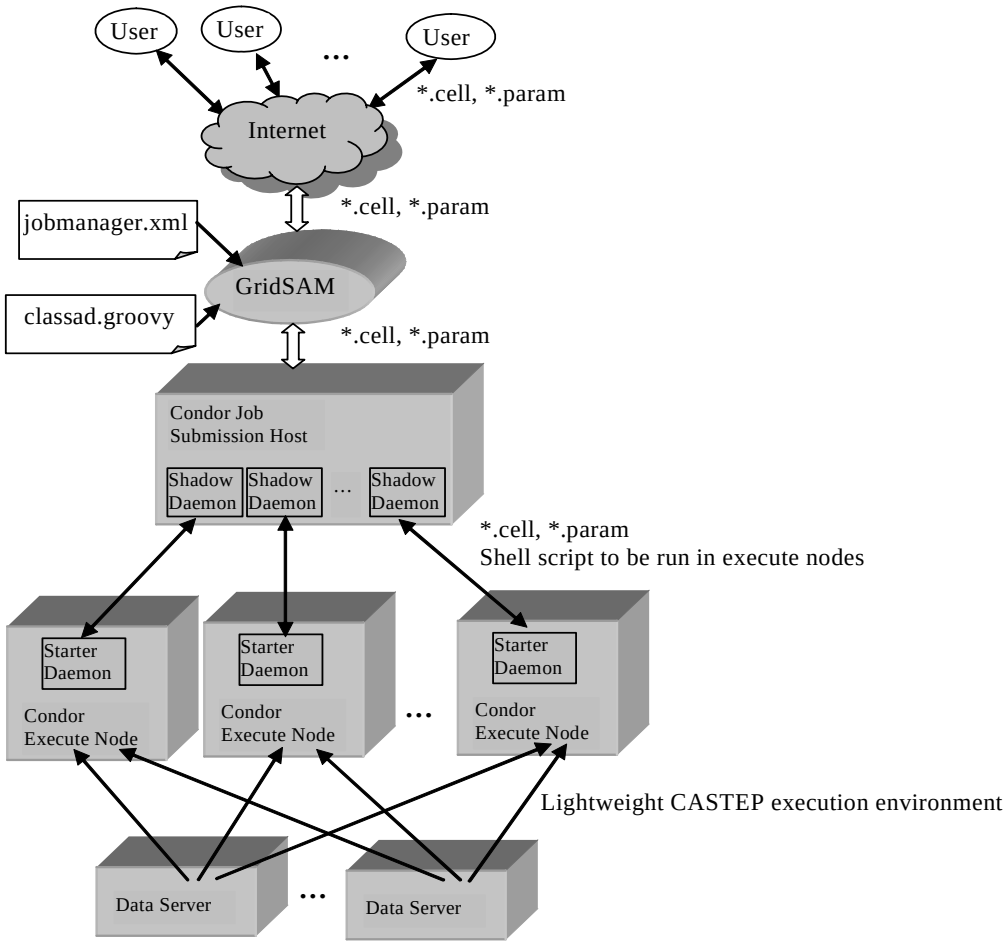


Figure 3. Application execution mechanism based on independent data servers

execution environment. These data servers can be any publicly accessible servers, such as FTP server, HTTP server, or more securely SFTP server. In the performance evaluation in Section 4 the FTP servers are utilised to offer file downloading. Multiple data servers can be set to provide the data replicas and the lightweight CASTEP environment can be downloaded from any one of them.

The access to data servers and execution of a CASTEP application are wrapped in an executable shell script. When the shell script is scheduled onto an execute node, it will first stage in the necessary files from one of the data servers and then call the downloaded CASTEP executable to perform the calculations. The Condor's file transfer mechanism is only used to transfer user-defined input files and this executable shell script. The shell script for the execution mechanism with 2 FTP servers is outlined in Fig.2. The submitted job request is to calculate the total energy of the Al_2O_3 crystal [5]. In the shell script, the files in step 1-2 contain the lightweight environment for running the CASTEP application, and the files in step 3-4 are the potential files needed for calculating the user-specified Al_2O_3 model. Step 5 generates a random number, which is 0 or 1 in this example. In step 6-13, a FTP server is specified according to the result of step 5. In step 14-22, the specified files are downloaded. Step 23-24 build the execution environment for running the CASTEP application. Step 25 invokes the shell script contained in the constructed execution environment to calculate the model.

The execution mechanism based on data servers is illustrated in Figure 3.

4. Performance Evaluation

This section evaluates the performance of the execution mechanism presented in this paper. Multiple users are simulated to simultaneously submit the CASTEP execution requests via GridSAM (each user submits one request) to calculate the total energy of the Al_2O_3 molecular model [5]. GridSAM then launches the batch of jobs through Condor onto CamGrid. GridSAM 1.1 [14] is deployed in the experiments. Please refer to [1] and [15]

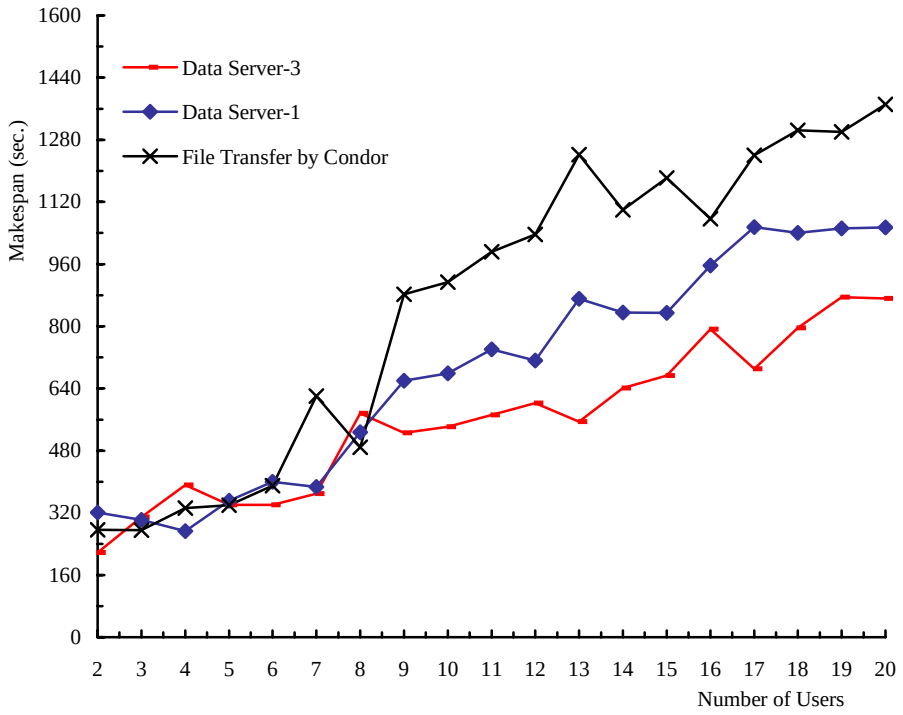


Figure 4. Performance comparison of different execution mechanisms in terms of makespan

for the detailed settings of CamGrid. In this paper, the experimental results for three mechanism settings are demonstrated. These three mechanism settings are:

1) Lightweight CASTEP execution environment is transferred through Condor's file transfer mechanism;
2) One FTP server is used to provide downloading of the lightweight CASTEP execution environment;
3) Three FTP servers are employed, i.e., three replicas are provided for the downloading. Each FTP sever offers the same network bandwidth.

4.1. Makespan

In this subsection, the experiments have been conducted to compare the performance of different execution mechanisms in terms of makespan as the number of the users submitting CASTEP jobs increases. Makespan is defined as the time duration between the time when GridSAM submits the batch of jobs to CamGrid and the time all these jobs are completed. The shorter makespan, the better performance is achieved. The experimental results are demonstrated in Fig.4.

It can be observed from this figure that when the number of users is small (less than 9) the performance curves under these three mechanisms are intertwined with each other in terms of makespan. As the number of the users increases to more than 9, the performance of these three cases differentiates from each other. The mechanism with 3 data servers outperforms the other two mechanisms while the mechanism which makes use of Condor's file transfer mechanism shows the worst performance.

These results can be explained as follows. When using Condor's file transfer mechanism, the execute nodes obtain the files from the job submission host. In the Condor system, the job submission host is a very busy node. It has to generate a Shadow daemon for every execute node it sends the job to. The Shadow daemon is responsible for communicating with the Starter daemon in each execute node, which includes transferring input files specified in the Condor submission description file. Therefore, although each execute node has a separate Shadow daemon in the job submission host for transferring files, these Shadow daemons work in the same node and share the same network card and a stretch of the network link, and as the result, these Shadow daemons may not be able to transfer the files in parallel, especially when their number becomes high.

Setting up an independent data server to offer data downloading can shift the burden of the job submission host. This can explain why the execution mechanism based on the data server outperforms that based on the Condor's file transfer mechanism in Fig.4. When there are more than one data servers to offer multiple data replicas, the file transfers can be processed in the higher

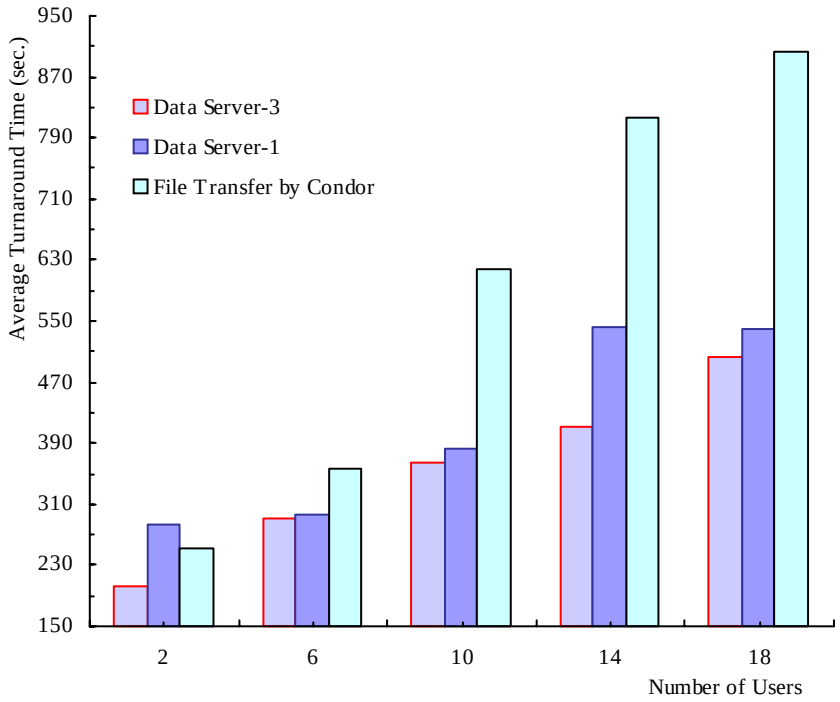


Figure 5. Performance comparison of different execution mechanisms in terms of average turnaround time

degree of parallelism. Hence, the mechanism with 3 data servers performs better than that with one data server. This result suggests that providing multiple data replicas for access are beneficial to improving performance in terms of makespan.

4.2. Average turnaround time

Fig.5 demonstrates the performance of different execution mechanism in terms of average turnaround time under different number of users. A job's turnaround time is defined as the time duration between the time when it is submitted and the time when the job is completed.

As can be observed from Fig.5, the execution mechanisms based on data servers significantly outperform the one based on the Condor's file transfer mechanism except the case of 2 users. This may be due to a smoother file transfer with independent data servers. It can also be observed from this figure that the mechanism with 3 data servers achieves better performance than the one with a single data server. Again, this can be explained by the fact that the files can be transferred in higher degree of parallelism when multiple data servers are accessible.

4.3 Overhead of the mechanisms

Fig.6 demonstrates the overhead of the mechanisms presented in this paper. The overhead imposed by using the mechanism to submit, schedule and run a CASTEP job is defined as all the time that is not spent on executing the job. The overhead includes the time for file transfer, queuing time, the time used to dispatch jobs as well as the time used to collect the calculated results. The average overhead of a batch of jobs is defined as the average of the overhead of all jobs. As can be seen from Fig.6, the data server-based execution mechanisms have much lower overhead than the one based on the Condor's file transfer mechanism when the number of users is greater than 2. This may be because that in the mechanism using Condor to transfer files, the job submission host becomes a bottleneck since the job management functionalities including file transfer are all performed on this node. In the mechanism based on data servers, a large portion of file transfer functionalities are shifted to data servers. Therefore, the job management can be carried out in the higher degree of parallelism so that the overhead endured by each job is likely to be reduced.

4.4 Speedup

The experiments conducted in this section investigate the speedup achieved by the CASTEP execution mechanisms presented in this paper. It is assumed that the CASTEP execution requests processed by the presented mechanisms are also sent to and run in sequence in the Materials Studio Server where the CASTEP execution

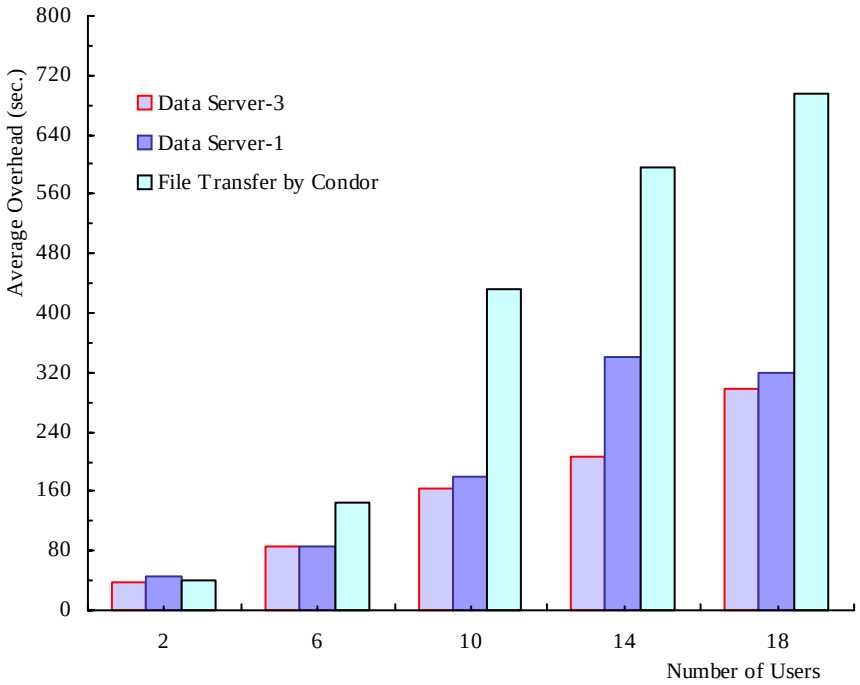


Figure 6. Average overhead of different mechanisms under different number of users

Table 1. Speedup of the CASTEP execution mechanism with 3 data servers in terms of makespan (time unit: second)

The number of users	2	6	10	14	18
Data Server-3	217.8	340.8	541.7	641.4	796.2
Materials Studio	331.4	1230.2	2017.0	2282.7	3590.2
Speedup	1.5	3.6	3.7	3.6	4.5

Table 2. Speedup of the CASTEP execution mechanism with a single data server in terms of makespan (time unit: second)

The number of users	2	6	10	14	18
Data server-1	320.9	399.7	679.0	835.0	1040.6
Materials Studio	472.4	1257.3	2030.3	2802.1	3908.4
Speedup	1.5	3.1	3.0	3.4	3.8

Table 3. Speedup of the CASTEP execution mechanism based on Condor's file transfer in terms of makespan (time unit: second)

The number of users	2	6	10	14	18
File transfer by Condor	276.2	390.2	913.3	1099.2	1304.3
Materials Studio	423.8	1273.0	1863.7	3093.5	3920.5
Speedup	1.5	3.2	2.4	2.8	3.0

environment is available (therefore, there is no need to transfer the lightweight execution environment).

The speedup of the execution mechanism in this paper over the Materials Studio can be defined as the ratio of the makespan achieved by the Materials Server to the makespan by the execution mechanism.

Table 1 shows the makespan achieved by the execution mechanism with 3 data servers, the makespan when the same batch of jobs is run in the Materials Studio, and the resulting speedup. Table 2 and Table 3 demonstrate the experimental results of the other two mechanisms in terms of speedup.

It can be seen from these tables, the speedup achieved by all these three mechanisms is greater than 1 even if there exist the overhead of transferring the lightweight CASTEP execution environment and the overhead imposed by using GridSAM job submission service and the Condor job manager. These overheads are relatively

constant for the given execution platform and execution mechanism. Therefore, if the molecular models with the longer calculation time are sent for execution, the speedup to be achieved is expected to be higher (the order of the time for calculating the Al_2O_3 molecular model in these experiments is about 200 seconds).

It can also be observed from these tables that the data server-based execution mechanisms achieve the higher speedup when the number of users is high (greater than 6). This result once again suggests that the data server-based execution mechanism is superior to the one based on the Condor's file transfer mechanism when the number of jobs submitted to the system is high.

5. Conclusions

CASTEP is an application using the density functional theory to calculate the structure and properties of materials and molecules. Currently, the Client/Server architecture in the Materials Studio is the popular approach to running CASTEP applications. This paper investigates the execution mechanisms for running CASTEP in grid resources, where CASTEP execution environments may not be available. In this paper, a lightweight CASTEP execution environment is developed and it can be transferred to any grid computational resource. Based on the lightweight environment, two execution mechanisms have been developed in this work. One makes use of the Condor's file transfer mechanism to transfer the lightweight environment and the other employs the independent data servers. The performance of these execution mechanisms is evaluated on CamGrid in terms of various metrics. The results show that the proposed execution mechanisms can speed up the batch processing of the CASTEP applications; moreover, the mechanisms based on data servers can achieve higher performance than the one based on Condor's file transfer mechanism, especially when the workload level is high. These execution mechanisms also have excellent potentials to be applied to other computational applications as long as the corresponding lightweight execution environments can be developed.

Acknowledgement

The authors would like to thank the DTI for sponsoring MaterialsGrid project.

6. References

[1] M Calleja, B Beckles, M Keegan, M A Hayes, A Parker, M T Dove, "CamGrid: Experiences in constructing a university-wide, Condor-based, grid at the University of Cambridge", *Proceedings of the 2004 UK e-Science All Hands Meeting*

[2] M Dove, E Artacho1, T White, R Bruin, et al, "The *e*Minerals project: developing the concept of the virtual organisation to support collaborative work on molecular-scale environmental simulations", *Proceedings of the 2005 UK e-Science All Hands Meeting*

[3] I. Foster and C. Kesselman, editors, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 2003, 2nd edition. ISBN: 1-55860-933-4

[4] C. Goonatilake, C. Chapman, W. Emmerich, M. Farrellee, T. Tannenbaum, M. Livny, M. Calleja and M. Dove, "*Condor Birdbath - web service interface to Condor*", *Proceedings of the 2005 UK e-Science All Hands Meeting*.

[5] V Hoang and S. Oh Molecular, "dynamics study of aging effects in supercooled Al_2O_3", *Phys. Rev. E*, 70, 061203, 2004

[6] W. Lee, A. McGough, and J. Darlington, "Performance Evaluation of the GridSAM Job Submission and Monitoring System", *Proceedings of the 2005 UK e-Science All Hands Meeting*, Nottingham, UK, 2005

[7] M.D. Segall, P.J.D. Lindan, M.J. Probert, C.J. Pickard, P.J. Hasnip, S.J. Clark and M.C. Payne, "First-principles simulation: ideas, illustrations and the CASTEP code", J. Phys. Condensed Matter 14 (2002) 2117

[8] D. Thain, T. Tannenbaum, and M. Livny, "Condor and the Grid", in Fran Berman, Anthony J.G. Hey, Geoffrey Fox, editors, *Grid Computing: Making The Global Infrastructure a Reality*, John Wiley, 2003. ISBN: 0-470-85319-0

[9] D. Thain, T. Tannenbaum, and M. Livny, "Distributed Computing in Practice: The Condor Experience" *Concurrency and Computation: Practice and Experience*, Vol. 17, No. 2-4, pages 323-356, February-April, 2005.

[10] J. Wakelin, A. García, P. Murray-Rust, "The use of XML and CML in Computational Chemistry and Physics Applications", *Proceedings of the 2004 UK e-Science All Hands Meeting*, 2004

[11] Yong Zhang, Peter Murray-Rust, Martin T Dove, Robert C Glen, Henry S Rzepa, Joa A Townsend, Simon Tyrell, Jon Wakelin, Egon L Willighagen, "JUMBO - An XML infrastructure for eScience", *Proceedings of the 2004 UK e-Science All Hands Meeting*.

[12] http://www.accelrys.com/products/mstudio/

[13] http://www.materialsgrid.org/

[14] http://gridsam.sourceforge.net/1.1/

[15] http://www.escience.cam.ac.uk/projects/camgrid/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

A Lightweight, Scriptable, Web-based Frontend to the SRB

Toby O. H. White[1], Rik P. Tyer[2], Richard P. Bruin[1], Martin T. Dove[1], Katrina F. Austen[1]

[1]Department of Earth Sciences, Downing Street, University of Cambridge. CB2 3EQ
[2]CCLRC Daresbury Laboratory, Daresbury, Warrington. WA4 4AD

Abstract

The San Diego Supercomputing Centre's Storage Resource Broker (SRB) is in wide use in our project. We found that none of the supplied interfaces fulfilled our needs, so we have developed a new interface, which we call TobysSRB. It is a web-based application with a radically simplified user interface, making it easy yet powerful to operate for novices and experts alike. The web interface is easily extensible, and external viewers may be incorporated. In addition, it has been designed such that interactions with TobysSRB are easily automatable, with a stable, well-defined and well-documented HTTP API, conforming to the REST (Representational State Transfer) philosophy. The focus has been on lightweightness, usability and scriptability.

Introduction

Introduction to the SRB

The Storage Resource Broker (SRB) is a distributed data storage product written and maintained by the San Diego Supercomputing Centre (SDSC)[1]. It is in wide use throughout the UK eScience community [2].

It is intended to allow users to access files, and collections of files seamlessly within a distributed environment. It provides an abstraction layer between the storage of data, which may be in multiple locations, on multiple filesystems, and the access of data, which is presented through a unified interface, transparent to the details of storage type or location.

An SRB system consists of four components:

- The MCAT (Metadata Catalogue) database, which stores internal SRB information - most importantly, mappings between a file's SRB address, and its storage location.
- The MCAT SRB server, which contains much of the logic for manipulating files and internal SRB state.
- The SRB server, which exports an interface across the network, accepting requests from clients, and translating these requests into interactions with the MCAT database and MCAT server.
- The SRB client, which is what the user or developer sees, and interacts with, and which exports the transparent, seamless, abstraction layer which is the point of the SRB.

Within the *e*Minerals project, we have been using the SRB for several years now as our primary data repository. It enables us to share data across the project in a very simple fashion. We have also built workflow tools using the SRB as a universally-accessible storage layer[3]. These tools are widely used across the project, and several large-scale studies have been performed using them.

SRB interfaces

From the user's perspective, regardless of the interface used, the SRB appears much like a distributed filesystem, with data in files, and files in collections that may be nested like directories.

Interfaces for users

SDSC provide three user-accessible front-ends to the SRB, through which end-users may interact with files, and navigate through the collections and datasets of the SRB:

- The Scommands, which are a series of command-line tools for Unix-like operating systems, written in C, which very roughly mimic native POSIX commands; thus to list the contents of a collection, `Sls` is used, where `ls` lists directory contents on a traditional Unix file system.
- MySRB, which is a web-based graphical interface.
- InQ, which is a native graphical MS Windows application.

In addition, there are a number of third-party user interfaces available, which all support subsets of the SRB's functionality[4].

Interfaces for developers

In addition, there are a number of officially supported SRB APIs available:

- There is a fully-featured C API, which is the primary developers' interface.
- Jargon is a Java API which exports the full range of functionality.
- Matrix exports a WS (Web Services) SRB interface, which is built on top of Jargon.

Again, there are also a number of third-party interfaces. Mostly these consist of language-specific wrappers around the C API.

Use of the SRB

The SRB's primary selling point is as a way to abstract access to files stored in multiple logical locations, through a single interface, which bears a resemblance to a hierarchical filesystem. In addition, it offers a number of additional features - limited user-editable metadata, and replication of files.

However, within our project, we have found that the only aspect of the SRB that we are interested in is the common view of a familiar filesystem-like interface.

This manner of usage is encouraged by the analogies that can be drawn between the Scommands and native unix filesystem tools.

Indeed, for both SRB versions 2 and 3, there are filesystem plugins, which enable the use of the SRB transparently, and allow an SRB repository to be mounted as a native filesystem within Linux, and a similar plugin to Windows Explorer, which enables an analogous interface for Windows.

Given that we use SRB only as a network filesystem, many of the features offered by the existing interfaces are beyond our needs.

Deficiencies in current methods of SRB interaction

User interfaces

The Scommands are the SRB developers' recommended interface. However, navigating an SRB repository through the Scommands has many of the same strengths and drawbacks as navigating a normal filesystem from the command line.

In its disfavour is the fact that visualizing a directory structure, and navigating through it when you are not sure of the destination, can be clumsy from the command line; and for users unused to command-line interaction, it is a daunting prospect. Indeed it is for this reason that graphical file system browsers were invented in the 1970s.

Furthermore, a network filesystem can never deliver the performance of a local filesystem, due to network latency. This is a familiar problem, from which all network filesystems (NFS, AFS) suffer. Thus Scommand-ing one's way through a repository is always slower and less efficient even than navigating through a filesystem using the analogous native unix tools.

A further deficiency is the necessity to install the Scommands on any system where access is required. It is impossible to connect to the SRB using this method from an arbitrary computer which knows nothing about the SRB. Furthermore, even when the Scommands are installed, it is necessary for each user individually to be set up correctly to use the SRB - thus it is not possible to, for example, lean over someone's shoulder, while they are logged on, and quickly retrieve a file from your own SRB collection.

In its favour is the fact that, since the interface is expressed and used through a unix shell, SRB interaction can be very easily incorporated in a script, whether written in shell script, or in a higher-level language such as Perl or Python.

However, again, network effects conspire to make repeated **`Sls`**'s a great deal slower than lots of **`ls`**'s. In addition, our experience has been that the SRB is not sufficiently robust to allow scripting of many SRB interactions (on the order of a few hundred in less than a minute). This is due to an architectural flaw in the SRB server, which will report success for a transaction, even before the database has been updated, thus breaking the atomicity of SRB operations. Clearly this introduces an enormous number of potential race conditions. It is therefore impossible to set up the SRB infrastructure to allow high volumes of requests of the sort which are essentially trivial for a real filesystem. Repeated failures of various sorts will be seen.

MySRB and InQ both try to solve some of the problems associated with the Scommands, in different ways.

InQ primarily tries to solve the problems associated with a textual interface. As a native Windows application, it must be installed locally, on a Windows computer. Thus, it does not solve the problems associated with needing to be at a particular computer. It does at least not require setting up for each user - it can be used to retrieve anyone's files from a single installation.

MySRB is a web-based interface to the SRB. It is a CGI script, implemented in C, which provides a stateful session of limited duration, and most SRB actions are possible through it. Since it is a web application, it is accessible from anywhere with a working web-browser and internet connection.

However, it suffers from a number of deficiencies in its user interface (UI). Firstly, due to the necessity to make available through MySRB almost all of the functionality of the SRB, the interface suffers from complexity and overcrowding.

Secondly, when using it as a simple method for retrieving files, there are two particular irritations which render it frustrating for the expert, and confusing for the novice.

- It second-guesses the browser's ability to show files - if a request is made to view a file of a type MySRB is not aware of, or indeed that MySRB thinks cannot be displayed by the browser, then MySRB will escape all HTML-sensitive characters, and insert the resultant translated file into an HTML frame between **`<PRE>`** tags. This naturally renders it impossible for the browser to render the output sensibly when retrieving an SVG file.
- When downloading a file, MySRB communicates with the browser such that every time, the browser's save dialogue tries to name the file **`MySRB.cgi`**.

Developer interfaces

In terms of scriptability, the available developers APIs did not fit our needs either.

Jargon is in Java, which is of course only useful for Java applications. Very few of our tools are written in Java; and Java is not conducive to quick scripting of the sort that the Scommands can be used for.

Matrix is a web-services API, and as such is nominally platform-independent. However, in practice, it requires a large investment in the stack of SOAP/WS infrastructure on any computer which must communicate with it. Again, there is no sense in which it can be used in a script-like fashion.

Of the third-party interfaces, none suited our purposes.

Motivation

A further problem with all the available interfaces, with the obvious exceptions of MySRB and Matrix, is that they all require communication between client and server over a number of uncommon ports; primarily port 5544 for MCAT communication, and variable ports from 65000 upwards. This makes use of the SRB from behind firewalls tricky. Clearly MySRB and Matrix work entirely over ports 80 or 443, which are almost universally available.

So, in summary, the main problems we perceived with the available methods of interacting with the SRB were:

I. The UIs of the available graphical approaches were insufficiently user-friendly
II. None of the interfaces available offered sufficiently robust scriptability.
III. Only the Scommands offered any scriptability at all, but required local installation and setup for each user, and could not be used from behind many firewalls.

To this end, we have built a new SRB frontend, which we call TobysSRB, which ameliorates all of these issues.

Approaches to a solution

The primary motivation for the creation of TobysSRB was to solve problem I above - we needed a very simple UI for retrieval of files from the SRB; its initial intended audience was in fact undergraduate students, who had little to no knowledge of the software involved, and who had little to no control over the computers from which they need to access the SRB.

The following requirements were thus essential.

- It should allow the very easy retrieval and viewing of files.
- It should not require any installation on client machines.

It was clear that a web-based solution would best fulfil these criteria, since web browsers are universally available, and a small amount of forethought combined with extensive user testing and feedback would ensure that the UI could be made sufficiently transparent that novice users would feel at ease, solving problem I above.

In addition, since we also perceived problems II and III above, we realised that, with proper design, a web-based solution could fix these also.

Problem II can only sensibly be solved by wrapping the Scommands - a non-robust, but scriptable, interface - behind a layer which performs proper error and timeout checking. This layer can as well be a web-based application as any other sort.

Problem III is of course solved in the same way - by making the interface web-accessible, it is then accessible anywhere.

Overview of TobysSRB

TobysSRB is implemented as a CGI script, written in Python, and all its interactions with the SRB are performed by executing appropriate Scommands using Python's subprocess handling.

For security, it is written such that it will only work over an HTTPS connection; the only information exposed to eavesdroppers is that a connection is being made; all data is encrypted.

It requires no special configuration on the server other than that required for running CGI scripts in general; and of course that the Scommands be installed somewhere on the server.

In this section, we shall explain firstly its internal implementation, then the interfaces presented to the user, both through a web-browser, and through its web-accessible API.

Internal implementation

Configuration and authentication

Configuration of the Scommands for a user involves the creation of a directory, **`~/.srb`**, and two files therein, **`~/.MdasEnv`** and **`~/.MdasAuth`**. Whenever the user wishes to interact with the SRB, they must first issue the command **`Sinit`**. This authenticates them against the server, and then creates a file within **`~/.srb`** which keeps the SRB session's state (which consists only of the location of the current SRB collection). The session should be ended with an **`Sexit`** which merely clears up this session file.

(Of course, frequently one will pause in the middle of a session, and forget to do an **`Sexit`** before logging out, with the result that the **`~/.srb`** directory quickly fills up with stale session files.)

```
mdasCollectionHome '/home/tow.eminerals'
mdasDomainHome     'eminerals'
srbUser            'tow'
srbHost            'forth.dl.ac.uk'
srbPort            '5544'
defaultResource    'CambsLake'
AUTH_SCHEME        'ENCRYPT1'
```

Figure 1: Example **`.MdasEnv`** File

A typical example of a **`.MdasEnv`** file is shown in figure 1. (The **`.MdasAuth`** file contains nothing but a password.) Note, however, that much of the contents is either redundant, or irrelevant to the client.

In almost all cases, the home directory can be constructed from the username - and in any case, the client ought not to need to specify their home directory when the server will also know. The client ought not to care about the authentication mechanism, when the server could tell it. Since there is a default port, it should not be necessary to specify it unless using a non-default value.

Thus, in fact the only information that the client should need to know is username, password, and location of the MCAT server. In our case, since TobysSRB is wrapping all SRB details, the client need not even know this last. All the user need specify to a given TobysSRB instance are username and password. All additional information is held by TobysSRB in a configuration file.

So in a given session with TobysSRB, the username and password are provided as CGI variables. TobysSRB then constructs a temporary directory, within which it constructs two files, corresponding to **`.MdasAuth`** and **`.MdasEnv`**. All necessary Scommands are then executed as follows:

```
MdasEnvFile=$TMPDIR/MdasEnvFile \
MdasAuthFile=$TMPDIR/MdasAuthFile \
Scommand
```

and at the end of a TobysSRB session, the files and the temporary directory are removed.

A brief note on password security: since all TobysSRB sessions occur over HTTPS, all information on passwords is secure from eavesdropping. However, it may be visible, as a CGI variable, in the URL bar of a web-browser. In order to obviate the possibility of password stealing by looking over shoulders, it is trivially obscured by firstly XORing the password character by character, and then encoding all URL-sensitive characters. This makes the password essentially immune to being picked up by a glance.

Session state

One of the main reasons why MySRB is difficult to script against is the fact that it is a stateful application, and cookie-handling is used to keep the session alive. This is illustrated in figure 2.

This has the advantage, of course, that the user is not required to reauthenticate every time they perform some action - rather the authentication data is held in a cookie. And since the authentication information required for MySRB is not merely username and password, but the full gamut of configuration information described in the previous section, it would be obnoxious to require typing in every time.

Unfortunately, of course, it also means that any client must be prepared to handle cookies to interact with MySRB, which effectively restricts clients to browsers, and excludes simple scripts. It also makes the internal workings of MySRB significantly more complicated, since session-handling logic is required.

However, TobysSRB works in an entirely stateless fashion, as shown in figure 3. This effectively means that reauthentication occurs on every action. However, this can be made transparent to the user - once the user has authenticated, every page that is returned from TobysSRB has a username/password form, but the values are filled in by default, so the user need not worry about them.

For a session consisting of multiple commands , this stateless approach theoretically involves an increase in load on the server side, since now a temporary directory and files must be created and destroyed for every request. (For single requests there is obviously no difference.) However, in practice, we have found this increase entirely unmeasurable. And from the client's perspective, any marginal increase in time is insignificant compared with the time required for each interaction between TobysSRB and the underlying SRB servers.

In addition, MySRB need only initialize the SRB session once for each MySRB session, whereas TobysSRB in principle must reinitialize every time. However, since TobysSRB is stateless, and the only purpose of **`Sinit`** is to set the current directory, in fact we need not initialize our session at all. If all SRB locations are specified as full (not relative) paths, then Scommands all work perfectly correctly without initialization; and this also obviates the need for us to worry about clearing up stale session files later on.

Furthermore, by keeping the implementation entirely stateless, it means that the interface is trivially scriptable, since no cookie-handling need be performed by the client.

Interaction with the SRB and error handling

As described, interaction with the SRB is performed by Scommands executed from within TobysSRB. Some notes on implementation here are worthwhile.

Firstly, since the Scommands are being executed, by the shell interpreter, it is vitally important that any input that is passed in from the user is checked before constructing the command line, otherwise malicious shell commands could be easily executed by the user. To this end, we check for safety all user input that will be passed to the command line, and escape any characters in SRB filenames that are also shell metacharacters.

This task is made more difficult by the fact that (prior to version 3.4.1 of the SRB) there is no definitive list of what characters are allowable in SRB filenames - indeed different official SRB tools allow different sets of characters, and files created through one tool may not be retrievable through another (*vide* frequent discussions on SRB-Chat); and the list of problematic characters varies between SRB versions. Therefore, we simply disallow any characters that might cause problems on any version of the SRB. This does prevent certain filenames, which are otherwise legal in recent SRB versions, from being used, but we feel our conservative approach is entirely justified.

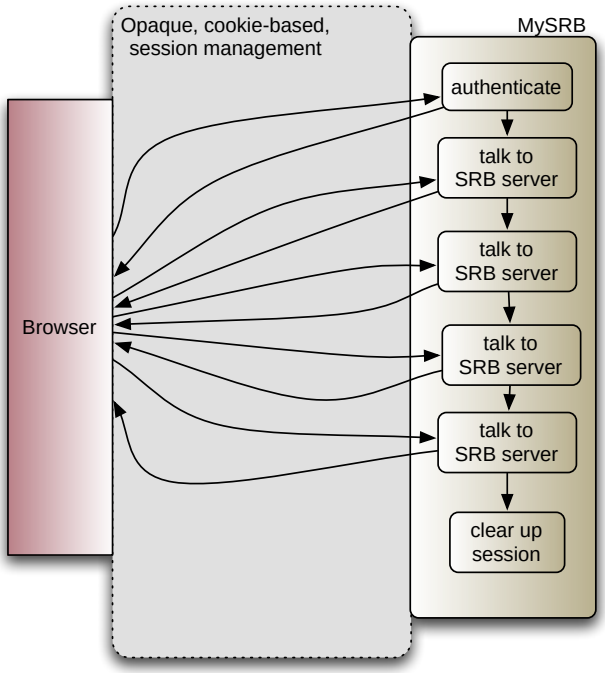


Figure 2: Illustration of MySRB session

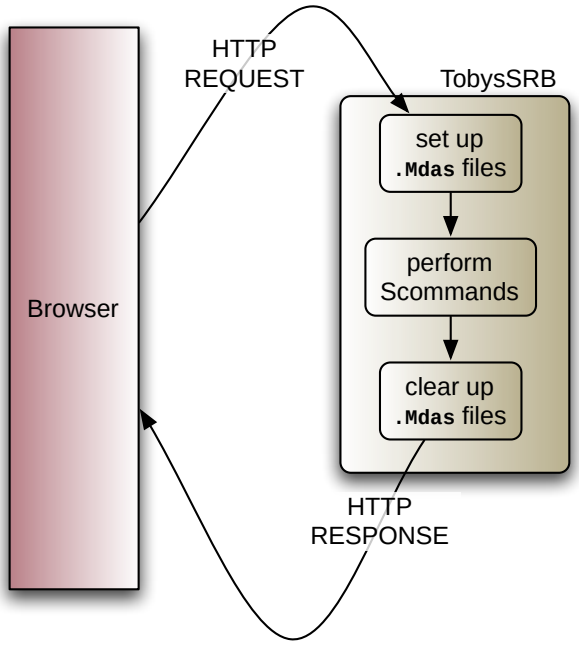


Figure 3: Illustration of TobysSRB session

Secondly, interactions with the SRB are never entirely robust. For example, occasionally an Scommand will hang indefinitely, or return with an obviously wrong error message.

This is excusable (or at least it is feasible to work with) when using the Scommands by hand - one can easily interrupt a hanging command; or re-execute one that has returned wrongly. However, when automating interactions, it is a major drawback, and thus TobysSRB is intended to wrap the Scommands and insulate the user against such vagaries.

Thus, each command is executed in a subprocess with a timeout, and TobysSRB repeatedly checks the status of each command issued; should the timeout be repeatedly exceeded, TobysSRB will return an appropriate error to the user.

Furthermore, the error handling of the Scommands is highly inconsistent - no documented scheme of error codes exists, so the cause of errors can only be deduced from reading the output of the commands; some of which are output on stdout, some on stderr; and there appears to be no pattern to their format. Furthermore, some error codes are overloaded, and the meaning of the error can only be deduced from the context of the request.

Therefore TobysSRB will also inspect both the stdout and stderr returned by the Scommands, and parse them to discover the cause of the error. Where the error appears to be of the type that is known to occur spuriously, the Scommand will be reissued a few times in the hope that it will succeed.

Finally, if the error occurs repeatedly, then TobysSRB will return to the user the error message, accompanied by an HTTP status code indicating the type of error.

As far as possible, TobysSRB will report error codes in accordance with the meanings assigned by HTTP/1.1[5]. Thus, 200 is only returned if the request was successful. If an upload was successfully performed the status is 201. If a request fails due to an authentication error, then the status is 401, while if the failure is due to an SRB timeout, the status is 408, and so on.

This enables TobysSRB to fit within the general framework of HTTP applications, and means any scripts written against it can deal with failures robustly and intelligently.

Extensibility

Because TobysSRB is written in well-constructed Python, with none of the complications associated with session management, it is a bare 500 lines long. Its control flow is thus easily grasped, and it is easily extended.

This was illustrated when a requirement arose to process XML files specially, by providing additional links to an external service which would transform the XML into a form more easily viewable in a web browser (described in further detail in [6]). It was a matter of ten extra lines of code to include this additional functionality.

Web application UI

For security, TobysSRB will only work over an HTTPS connection; if accessed over unencrypted HTTP, it will refuse to grant access, and try to redirect the browser to an appropriate HTTPS address.

On first accessing TobysSRB, the user is asked for a username and password, from which the location of the user's home collection is established, and a listing of the contents of that collection is

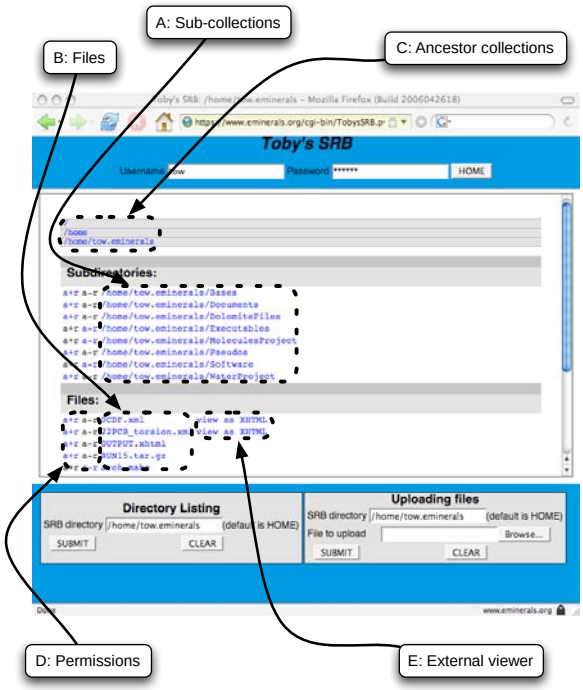


Figure 4: Screenshot of TobysSRB

retrieved and presented to the user. A screenshot is shown in figure 4.

As can be seen, collections (A) and individual data objects (B) are listed separately. The collection names are links which will return a page of the same format, displaying the contents of that collection.

The filenames are all links which will directly return the files; thus following them will allow the browser to render them however it can, and using the normal browser mechanism (usually right-click and select “Save As...”) will allow downloading the file. By constructing the URL appropriately, we have ensured that when saving the file, the browser knows the filename and will save it under that name.

In the top left of the screen, there is a list (C) of all the parent collections, which allows quickly navigating back up the SRB.

At the bottom of the screen are two forms; the first allows for quickly listing a given directory rather than having to navigate page-by-page throughout the hierarchy; the second allows uploading local files.

Two further things are worthy of note. Firstly, notice that each collection or filename is preceded by two links (D) labelled '**a+r**' and '**a-r**'. These change the permissions on the file or collection (recursively for collections) and add and remove, respectively, global read permissions.

The SRB of course offers much greater granularity of permissions, but we have found that the only changes we generally make are to global read permissions, and a quick button click is distinctly easier than **Schmod**'s arcane and poorly documented syntax. Trying to allow for the full range of allowable permission modifications would significantly complicate the interface.

Finally, as previously mentioned, all files suffixed **.xml** may have an additional link appended (E). This is because most of the XML files we produce within our project are CML files, and we have also developed an on-the-fly CML-to-XHTML convertor to allow easy viewing of such files[6].

Scriptable API

Part of the purpose in creating TobysSRB was to provide a stable and robust programmable interface to the SRB, accessible from computers where the Scommands are not installed. For this reason, all TobysSRB functions are accessible with a simple and well-documented API, described in this section.

TobysSRB receives information from a client on two channels;

- HTTP verbs
- CGI argument list.

HTTP verb

TobysSRB understands the following four HTTP verbs, and performs appropriate actions:

- GET - for retrieval of information; either of a directory listing or of file contents.
- PUT - a file will be uploaded.
- DELETE - a file or collection will be deleted.
- POST - one of a variety of other actions may be taken.

When TobysSRB is invoked from a web-browser, only GET and POST will be used; but PUT and DELETE can be used from scripts which issue HTTP requests.

CGI argument list

A range of CGI arguments are recognized, which TobysSRB will act upon. A full description of the API is beyond the scope of this paper, but by appropriate combinations of parameters, all of the SRB actions TobysSRB knows about may be performed.

The API is clearly documented, and easily understood. Most importantly, files are easily accessible from a single, stable, easily generated URL, which looks like

```
https://my.server.ac.uk/TobysSRB.py/
path/to/filename?
username=user;password=pass;
```

By accessing that URL with GET, the object may be retrieved from anywhere, accessing it with PUT will place data into the file, accessing it with DELETE will remove the file, and - in concert with additional CGI parameters - accessing it with POST will perform all other available operations.

If the URL ends in a ‘/’ then it is assumed that the relevant SRB object is a collection, and so a listing of its contents will be returned; if not then it is assumed to be a data object, whose contents will be retrieved and returned. If either assumption is wrong, a redirect will be issued, in the same way as would happen for an HTTP request.

Philosophy

This API very much follows the REST (Representational State Transfer) philosophy [7] and may be seen as a lightweight alternative to wrapping the SRB with Web Services. Each SRB collection and file may be perceived, through TobysSRB, as an HTTP-accessible resource, upon which various operations (retrieval, modification, deletion, *etc*.) may be performed.

In this fashion, all of the SRB operations supported by TobysSRB may be used from a script capable of running on any internet-connected machine, without recourse to the Scommands.

Thus the command

```
Sget /path/to/file_of_interest
```

may be replaced, when Scommands are unavailable, with

```
wget "https://my.server.ac.uk/
TobysSRB.py/path/to/file
username=user;password=pass"
```

Other interactions are easily performed by generating more complex HTTP requests, some of which can be done with **`wget`** or **`curl`** (which are almost universally available on the Unix command line), and for those which cannot, generation of an HTTP request is less than 10 lines of easily abstractable Perl or Python since modules exist for this in the standard libraries of both.

It should be noted that the SRB username and password must be known by the script in order to create the URLs. In simple cases, they could be stored inline in the script, but they could equally be dynamically read from a file, or generated by user input. This does not however make the method any less secure than other existing methods the Scommands need to know these data as well, but simply store them in the **`.srb`** directory. By allowing them to be stored or generated elsewhere, we place more power in the hands of the user, since they do not need to create **`.srb`** directories on every machine; nor are the usernames and passwords so easily discoverable should a client account be compromised.

This API therefore provides an accessible way of automating SRB interactions. In addition, since TobysSRB acts as a buffering layer against the vagaries of the SRB, and returns well-documented error messages, it can act as a considerably more robust SRB interface for scripts, even where the Scommands are available. Since no additional client software is necessary, it can be immediately used on any internet-connected machine without preparation. And finally, the interaction occurs wholly over a single port, which generally speaking will be port 443, and universally available, so the resultant scripts are portable to any client machine without the need to worry about firewall issues.

Comparison with MySRB

Clearly, in some respects, TobysSRB is a direct replacement for MySRB, as a web-based interface to the SRB. A brief comparison follows.

MySRB supports a much wider range of SRB operations. It is the primary interface where new features in the C API are prototyped and exposed to the user.

In comparison, TobysSRB supports only the operations

- file retrieval
- file upload
- file delete
- collection listing
- collection creation
- collection removal
- add/remove global read permissions

However, in our experience, these compose by far the vast majority of operations performed, and are the only ones that we have found it useful to script - in any case, all other operations remain available through the Scommands.

Because TobysSRB supports a much reduced range of operations, its UI can be much simpler. This means that it could be designed in a fashion analogous to a typical graphical filesystem browser, which makes it easily accessible to any computer user familiar with that paradigm.

In addition, MySRB makes a distinction between viewing a file and retrieving it - when viewing it, MySRB second-guesses the browsers rendering capabilities, and massages the output in various ways before rendering it in a frame. This means that, for example, an SVG file may not be viewed through MySRB, because it is transformed into fully escaped HTML and presented to the browser as text.

TobysSRB, on the other hand, presents the file straight to the browser, mime-typed accordingly, and allows the browser to render the file how it, or the user, chooses.

Further, since MySRB is a stateful application, which uses cookies for session handling, it is very complicated to automate a MySRB session.

TobysSRB, however, works entirely statelessly, and therefore scripting an upload or download is as simple as performing (through **`curl`**, or **`wget`**, or Perl or Python) an HTTP request to a URL.

Finally, although the source for MySRB is available, so in principle it could have been altered in order to fulfil our requirements, and would be extensible for the addition of external links, in practice this is not the case, since it consists of 18000 lines of code, which is somewhat opaque to the uninitiated. In contrast, TobysSRB consists of 500 lines of Python, and is much more easily altered.

User experiences

TobysSRB is now in use across the (multi-institutional) *e*Minerals project which employs the authors, both for project members, and for the undergraduate students who work with the project. In addition it has been disseminated to a number of other users within the institutions hosting *e*Minerals. To the best of the authors' knowledge, every user who has been exposed to TobysSRB prefers it to MySRB.

Some users, especially the undergraduate students at whom it was initially aimed, use TobysSRB as their only method of SRB interaction. Other more advanced users continue to use the Scommands for some, if not most tasks, but when web access is required (for use from remote computers) or a browser interface preferred (for viewing XHTML or SVG files), then TobysSRB is unanimously preferred to MySRB.

The scriptable interface has not yet been as widely adopted, largely because there are a number of existing tools which already use the Scommands as their primary interface. However, several users do prefer the TobysSRB API, and a growing number of newer scripts are being written to work with that interface.

Summary

Finding the currently available methods for interaction with the SRB to be inadequate for our needs, a new front-end was developed. This interface, TobysSRB, pares down the facilities offered by MySRB and in so doing allows for the generation of a considerably more user-friendly interface.

TobysSRB allows the viewing of any SRB file, and is easily extensible such that it can, for example, automatically create links to external services, such as the CML viewing tool described in [6]. The primary objective of TobysSRB was to provide users within the *e*Minerals project with a user- friendly interface that performs all the commonly required tasks with greater facility than those tools already available.

Furthermore, the requirement for intelligent error-detection and timeout handling when wrapping the Scommands has resulted in the creation of a *REST*ful HTTP API for interacting with the SRB, allowing SRB interaction in a robust and network-transparent, universally accessible fashion.

Both objectives have been achieved, and TobysSRB has become a transferable tool that can be used by any project using the SRB.

Acknowledgements

We are grateful for funding from NERC (grant reference numbers NER/T/S/2001/00855, NE/C515698/1 and NE/C515704/1).

References

[1] Moore, RW and Baru, C., *"Virtualization services for data grids"* in *"Grid Computing: Making the Global Infrastructure a Reality"* (ed. Berman, F. Hey, A.J.G and Fox, G., John Wiley) Chapter 11 (2003);
Also, see http://www.sdsc.edu/srb

[2] Doherty, M., *et al, "SRB in Action"*, All Hands Meeting, Nottingham, 2003;
Manandhar, A. S. *et al."Deploying a distributed data storage system in the UK National Grid Service using federated SRB"*, All Hands Meeting, Nottingham, 2004;
Berrisford, P., et al., *"SRB in a Production Context"*, All Hands Meeting, Nottingham, 2004

[3] Bruin, R.P., et al. *"Job submission to grid computing environments"*, All Hands Meeting, Nottingham (2006) - in press;
Calleja, M. et al. *"Grid Tool integration with the eMinerals project"*, All Hands Meeting, Nottingham (2005);
Calleja, M. *et al., "Collaborative grid infrastructure for molecular simulations: the eMinerals minigrid as a prototype integrated compite and data grid"*, Mol. Simul. **31**, 303 (2005)
Chapman, C. *et al., "Managing Scientific Processes on the eMinerals mini-grid using BPEL"*, All Hands Meeting, Nottingham (2006) - in press

[4] http://www.sdsc.edu/srb/index.php/Contributed_Software

[5] Fielding, R. *et al., "Hypertext Transfer Protocol -- HTTP/1.1"*, RFC 2616, 1999.

[6] White, T.O.H. *et al., "Application and Use of CML in the eMinerals project"*, All Hands Meeting, Nottingham, 2006.

[5] Fielding, R. *et al., "Hypertext Transfer Protocol -- HTTP/1.1"*, RFC 2616, 1999.

[7] Fielding, R., *"Architectural Styles and the Design of Network-based Software Architectures"*, PhD thesis, University of California Irvine, 2000.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

A Lightweight Application Hosting Environment for Grid Computing

P. V. Coveney, S. K. Sadiq, R. S. Saksena, M. Thyveetil, and **S. J. Zasada**

Centre for Computational Science, Department of Chemistry,
University College London, Christopher Ingold Laboratories,
20 Gordon Street, London, WC1H 0AJ

M. Mc Keown, and S. Pickles

Manchester Computing, Kilburn Building,
The University of Manchester, Oxford Road,
Manchester, M13 9PL

Abstract

Current grid computing [1, 2] technologies have often been seen as being too heavyweight and unwieldy from a client perspective, requiring complicated installation and configuration steps to be taken that are beyond the ability of most end users. This has led many of the people who would benefit most from grid technology, namely application scientists, to avoid using it. In response to this we have developed the Application Hosting Environment, a lightweight, easily deployable environment designed to allow the scientist to quickly and easily run unmodified applications on remote grid resources. We do this by building a layer of middleware on top of existing technologies such as Globus, and expose the functionally as web services using the WSRF::Lite toolkit to manage the running application's state. The scientist can start and manage the application he wants to use via these services, with the extra layer of middleware abstracting the details of the particular underlying grid middleware in use. The resulting system provides a great deal of flexibility, allowing clients to be developed for a range of devices from PDAs to desktop machines, and command line clients which can be scripted to produce complicated application workflows.

I Introduction

We define grid computing as distributed computing conducted transparently by disparate organisations across multiple administrative domains. Fundamental to the inter-institutional sharing of resources in a grid is the grid middleware, that is the software that allows the institution to share their resources in a seamless and uniform way.

While many strides have been made in the field of grid middleware technology, such as [3, 4], the prospect of a heterogeneous, on-demand computational grid as ubiquitous as the electrical power grid is still a long way off. Part of the problem has been the difficulty to the end user of deploying and using many of the current middleware solutions, which has lead to reluctance amongst some researchers to actively embrace grid technology [5].

Many of the current problematic grid middleware solutions can be characterised as what we define as 'heavyweight', that is they display some or all of the following features:

i. the client software is difficult to configure or install, very often requiring an experienced system administrator to do so.

ii. they are dependent on lots of supporting software being installed, particularly libraries that are not likely to already be installed on the resource, or modified versions of common libraries.

iii. they require non-standard ports to be opened on firewall, requiring the intervention of a network administrator.

iv. they have a high barrier to entry, meaning that potential users have to develop a new skill set before they are able to use the technology productively.

To address these deficiencies there is now much attention focused on 'lightweight' middleware solutions such as [6] which attempt to lower the barrier of entry for users of the grid.

II The Application Hosting Environment

In response to the issues raised above we have developed the Application Hosting Environment (AHE), a lightweight, WSRF [7] compliant, web services based environment for hosting scientific applications on the grid. The AHE

allows scientists to quickly and easily run unmodified, legacy applications on grid resources, managing the transfer of files to and from the grid resource and allowing the user to monitor the status of the application. The philosophy of the AHE is based on the fact that very often a group of researchers will all want to access the same application, but not all of them will possess the skill or inclination to install the application on a remote grid resource. In the AHE, an expert user installs the application and configures the AHE server, so that all participating users can share the same application. This draws a parallel with many different communities that use parallel applications on high performance compute resources, such as the UK Collaborative Computational Projects (CCPs) [8], where a group of expert users/developers develop a code, which they then share with the end user community. In the AHE model, once the expert user has configured the AHE to share their application, end users can use clients installed on their desktop workstations to launch and monitor the application across a variety of different computational resources.

The AHE focuses on applications not jobs, with the application instance being the central entity. We define an application as an entity that can be composed of multiple computational jobs; examples of applications are (a) a simulation that consists of two coupled models which may require two jobs to instantiate it and (b) a steerable simulation that requires both the simulation code itself and a steering web service to be instantiated. Currently the AHE has a one to one relationship between applications and jobs, but this restriction will be removed in a future release once we have more experience in applying these concepts to scenarios (a) and (b) detailed above.

III Design Considerations

The problems associated with 'heavyweight' middleware solutions described above have greatly influenced the design of the Application Hosting Environment. Specifically, they have led to the following constraints on the AHE design:

- the user's machine does not have to have client software installed to talk directly to the middleware on the target grid resource. Instead the AHE client provides a uniform interface to multiple grid middlewares.
- the client machine is behind a firewall that uses network address translation (NAT) [27]. The client cannot therefore accept inbound network connections, and has to poll the AHE server to find the status of an application instance.
- the client machine needs to be able to upload input files to and download output files from a grid resource, but does not have GridFTP client software installed. An intermediate file staging area is therefore used to stage files between the client and the target grid resource.
- the client has no knowledge of the location of the application it wants to run on the target grid resource, and it maintains no information on specific environment variables that must be set to run the application. All information about an application and its environment is maintained on the AHE server.
- the client should not be affected by changes to a remote grid resource, such as if its underlying middleware changes from GT2 to GT4. Since GridSAM is used to provide an interface to the target grid resource, a change to the underlying middleware used on the resource doesn't matter, as long as it is supported by GridSAM.
- the client doesn't have to be installed on a single machine; the user can move between clients on different machines and access the applications that they have launched. The user can even use a combination of different clients, for example using a command line client to launch an application and a GUI client to monitor it. The client therefore must maintain no information about a running application's state. All state information is maintained as a central service that is queried by the client.

These constraints have led to the design of a lightweight client for the AHE, which is simple to install and doesn't require the user to install any extra libraries or software. The client is required to launch and monitor application instances, and stage files to and from a grid resource. The AHE server must provide an interface for the client to launch applications, and must store the state of application instances centrally. It should be noted that this design doesn't remove the need for middleware solutions such as Globus on the target grid resource; indeed we provide an interface to run applications on several different underlying grid middlewares so it is essential that grid resource providers maintain a

supported middleware installation on their machines. What the design does do is simplify the experience of the end user.

Communication in the AHE is secured using Transport Layer Security (TLS) [28]; our initial analysis showed that we did not need to use SOAP Message Level Security (MLS) as our SOAP messages would not need to pass through intermediate message processing steps. TLS is provided by mutually authenticate SSL between the client and the AHE, and between the AHE and GridSAM. This requires that the AHE server and associated GridSAM instances have X.509 certificates supplied by a trusted certificate authority (CA), as do any users connecting to the AHE. When using the AHE to access a computational grid, typically both user and server certificates will be supplied by the grid CA that the user is submitting to. Where proxy certificates are required, for example when using GridSAM to submit jobs via Globus, a MyProxy server is used to store proxy certificates uploaded by the user, which are retrieved by GridSAM in order to submit the job on the user's behalf.

IV Architecture of the AHE

The AHE represents an application instance as a stateful WS-Resource[7], the properties of which include the application instance's name, status, input and output files and the target grid resource that the application has been launched on. Details of how to launch the application are maintained on a central service, in order to reduce the complexity of the AHE client.

The design of the AHE has been greatly influenced by WEDS (WSRF-based Environment for Distributed Simulations)[9], a hosting environment designed for operation primarily within a single administrative domain. The AHE differs in that it is designed to operate across multiple administrative domains seamlessly, but can also be used to provide a uniform interface to applications deployed on both local HPC machines, and remote grid resources.

The AHE is based on a number of pre-existing grid technologies, principally GridSAM [10] and WSRF::Lite [11]. WSRF::Lite is a Perl implementation of the OASIS Web Services Resource Framework specification. It is built using the Perl SOAP::Lite [12] web services toolkit, from which it derives its name. WSRF::Lite provides support for WS-Addressing [13], WS-ResourceProperties [14], WS-ResourceLifetime [15], WS-ServiceGroup [16] and WS-BaseFaults [17]. It also provides support for digitally signing SOAP [18] messages using X.509 digital certificates in accordance with the OASIS WS-Security [19] standard as described in [20].

GridSAM provides a web services interface for submitting and monitoring computational jobs managed by a variety of Distributed Resource Managers (DRM), including Globus [3], Condor [21] and Sun Grid Engine [22], and runs in an OMII [23] web services container. Jobs submitted to GridSAM are described using Job Submission Description Language (JSDL) [24]. GridSAM uses this description to submit a job to a local resource, and has a plug-in architecture that allows adapters to be written for different types of resource manager. In contrast to WEDS, which represents jobs co-located on the hosting resource, the AHE can submit jobs to any resource manager for which a GridSAM plug-in exists.

Reflecting the flexible philosophy and nature of Perl, WSRF::Lite allows the developer to host WS-Resources in a variety of ways, for instance using the Apache web server or using a standalone WSRF::Lite Container. The AHE has been designed to run in the Apache [25] container, and has also been successfully deployed in a modified Tomcat [26] container.

Figure 1 shows the architecture and workflow of the AHE. Briefly, the core components of the AHE are: the App Server Registry, a registry of applications hosted in the AHE; the App Server Factory, a "factory" according to the Factory pattern [29] used to produce a WS-Resource (the App WS-Resource) that acts as a representation of the instance of the executing application. The App Server Factory is itself a WSRF WS-Resource that supports the WS-ResourceProperties operations. The Application Registry is a registry of previously created App WS-Resources, which the user can query to find previously launched application instances. The File Staging Service is a WebDAV [30] file server which acts as an intermediate staging step for application input files from the user's machine to the remote grid resource. We define the staging of files to the File Staging Service as "pass by value", where the file is transferred from the user's local machine to the File Stage Service. The AHE also supports "pass by reference", where the client supplies a URI to file required by the application. The MyProxy Server is used to store proxy credentials required by GridSAM to submit to Globus job queues. As described above we use GridSAM to provide a web services compliant front end to remote grid resources.

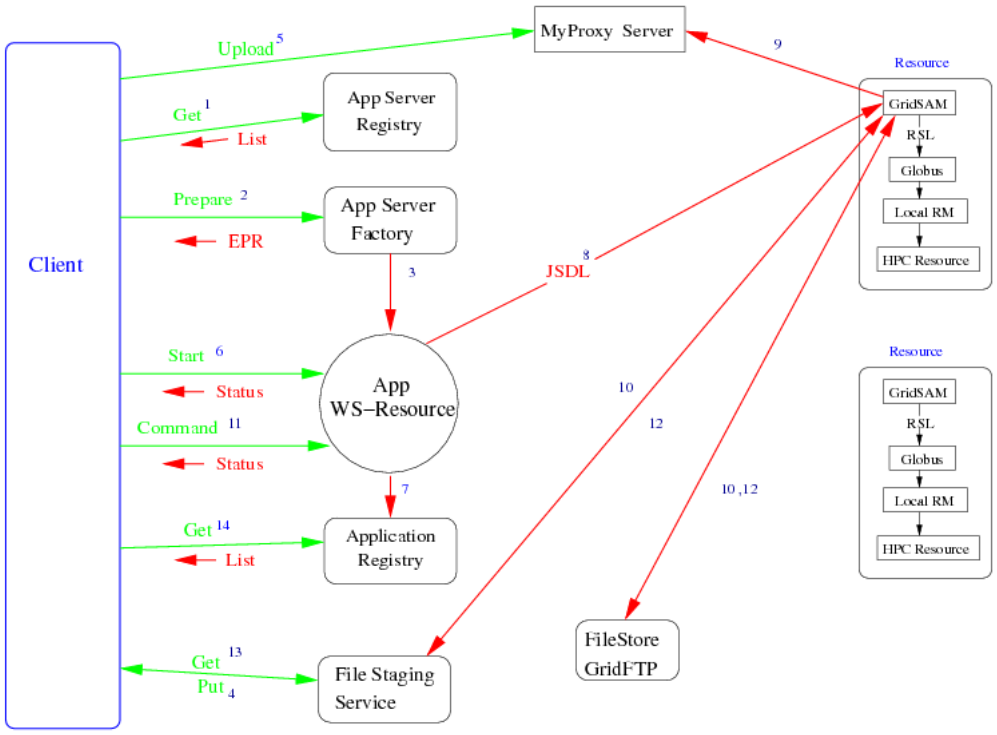


Figure 1: The architecture of the Application Hosting Environment

All user interaction is via a client that communicates with the AHE using SOAP messages. The workflow of launching an application on a grid resource running the Globus middleware (shown in figure 1 is as follows: the user retrieves a list of App Server Factory URIs from the AHE (1). There is an application server for each application configured in the AHE. This step is optional as the user may have already cached the URI of the App Server Factories he wants to use. The user issues a "Prepare" message (2); this causes an App WS-Resource to be created (3) which represents this instance of the application's execution. To start an application instance the user goes through the sequence: Prepare →Upload Input Files →Start, where Start actually causes the application to start executing. Next the user uploads the input files to the intermediate file staging service using the WebDAV protocol (4).

The user generates and uploads a proxy credential to the MyProxy server (5). The proxy credential is generated from the X.509 certificate issued by the user's grid certificate authority. This step is optional, as the user may have previously uploaded a credential that is still valid. Once the user has uploaded all of the input files he sends the "Start" message to the App WS-Resource to start the application running (6). The Start message contains the locations of the files to be staged in to and out from the target grid resource, along with details of the user's proxy credential and any arguments that the user wishes to pass to the application. The App WS-Resource maintains a registry of instantiated applications. Issuing a prepare message causes a new entry to be added to the registry (7). A "Destroy" command sent to the App WS-Resource causes the corresponding entry to be removed from the registry.

The App WS-Resource creates a JSDL document for a specific application instance, using its configuration file to determine where the application is located on the resource. The JSDL is sent to the GridSAM instance acting as interface to the grid resource (8), and GridSAM handles authentication using the user's proxy certificate. GridSAM retrieves the user's proxy credential from the MyProxy server (9) which it uses to transfer any input files required to run the application from the intermediate File Staging Service to the grid resource (10), and to actually

submit the job to a Globus back-end.

The user can send Command messages to the App WS-Resource to monitor the application instance's progress (11); for example the user can send a "Monitor" message to check on the application's status. The App WS-Resource queries the GridSAM instance on behalf of the user to update state information. The user can also send "Terminate" and "Destroy" messages to halt the application's execution and destroy the App WS-Resource respectively. GridSAM submits the job to the target grid resource and the job completes. GridSAM then moves the output files back to the file staging locations that were specified in the JSDL document (12). Once the job is complete the user can retrieve any output files from the application from the File Staging Service to their local machine. The user can also query the Application Registry to find the end point references of jobs that have been previously prepared (14).

V AHE Deployment

As described above the AHE is implemented as a client/server model. The client is designed to be easily deployed by an end user, without having to install any supporting software. The server is designed to be deployed and configured by an expert user, who installs and configures applications on behalf of other users.

Due to the reliance on WSRF::Lite, the AHE server is developed in Perl, and is hosted in a container such as Apache or Tomcat. The actual AHE services are an ensemble of Perl scripts that are deployed as CGI scripts in the hosting container. To install the AHE server, the expert user must download the AHE package and configure their container appropriately. The AHE server uses a PostgreSQL [31] database to store the state information of the App WS-Resources, which must also be configured by the expert user. We assume that a GridSAM instance has been configured for each resource that the AHE can submit to.

To host an application in the AHE, the expert user must first install and configure it on the target grid resource. The expert user then configures the location and settings of the application on the AHE server and creates a JSDL template document for the application and the resource. This can be done by cloning a pre-existing JSDL template. To complete the installation the expert user runs a script to repopulate the Application Server Registry; the AHE can be updated dynamically and doesn't require restarting when a new application is added.

The AHE is designed to be interacted with by a variety of different clients. The clients we have developed are implemented in Java using the Apache Axis [32] web services toolkit. We have developed both GUI and command line clients from the same Java codebase. The GUI client uses a wizard to guide a user through the steps of starting their application instance. The wizard allows users to specify constraints for the application, such as the number of processors to use, choose a target grid resource to run their application on, stage all required input files to the grid resource, specify any extra arguments for the simulation, and set it running.

To install the AHE clients all an end user need do is download and extract the client, load their X.509 certificate into a Java keystore using a provided script and set an environment variable to point to the location of the clients. The user also has to configure their client with the endpoints of the App Server Registry and Application Registry, and the URL of their file staging service, all supplied by their AHE server administrator.

The AHE client attempts to discover which files need to be staged to and from the resource by parsing the application's configuration file. It features a plug-in architecture which allows new configuration file parsers to be developed for any application that is to be hosted in the AHE. The parser will also rewrite the user's application configuration file, removing any relative paths, so that the application can be run on the target grid resource. If no plug-in is available for a certain application, then the user can specify input and output files manually.

Once an application instance has been prepare and submitted, the AHE GUI client allows the user to monitor the state of the application by polling its associated App WS-Resource. After the application has finished, the user can stage the application's output files back to their local machine using the GUI client. The client also gives the user the ability to terminate an application while it is running on a grid resource, and destroy an application instance, removing it from the AHE's application registry. In addition to the GUI client a set of command line clients are available which provide the same functionality of the GUI. The command line clients have the advantage that they can be called from a script to produce complex workflows with multiple application executions.

VI User Experience

We have successfully used the AHE to deploy two parallel molecular dynamics codes, LAMMPS [33] and NAMD [34]. These applications have been used to conduct production simulations on both the UK National Grid Service (NGS) and the US TeraGrid. There follows a discussion of two different use cases where the AHE has been used to quickly and easily run simulations using the grid.

A NAMD

Users often require the ability to launch multiple instances of the same or similar simulations that vary in particular attributes that affect the outcome of the simulation. An example of this is 'ensemble' molecular dynamics simulations of biological molecules in which the starting energies of various atoms are randomized to allow for conformational sampling of the biological structure through multiple simulations. Another example is Thermodynamic Integration (TI) techniques that calculate binding affinities between biological molecules. Given that enough grid resources are available, multiple jobs each utilizing a slightly different configuration can be launched and executed simultaneously to provide the necessary results. Prior to the AHE, the problems with implementing such techniques have been the tediousness of repetitive job submission coupled with the monitoring of job status across multiple grid resources, as well as the time consuming act of shepherding input and output files around from resource to resource.

The AHE circumvents these problems by presenting a uniform interface to multiple resources, through which multiple job submission can be achieved by scripting the AHE command line clients, as well as the ability to monitor each job through this interface. Furthermore, all files required for a job can be automatically staged to a set of desired resources as well as output files retrieved upon job completion.

Some molecular dynamics simulations also require complex equilibration protocols that evolve a biological molecule from an available starting structure to an equilibrium state at which relevant data can be collated. Such protocols usually involve a series of chained simulations where the output of one simulation is fed into the input of the next. Whilst some conventional methods such as ssh can be employed to afford some automation of chained job submission, scripting the AHE command line clients provides a simpler and quicker mechanism through which chaining can be distributed seamlessly across multiple grid resources.

B LAMMPS

The microscopic and macroscopic behaviour of large-scale anionic and cationic clay nanocomposite systems can be modeled using molecular dynamics (MD) techniques. The use of computer simulations to model these sort of systems has proved to be an essential adjunct to experimental techniques [35]. The clay systems which we simulate are those of the smectite clay, montmorillonite and the layered double hydroxide, hydrotalcite. Clays such as these form a sheet-like (layered) structure, which can intercalate molecules within their layers. Whilst useful information about the intercalated species can be obtained by running small-scale simulation, finite size effects can be explored by increasing the model size.

LAMMPS is an example of a well used MD code which does not have the functionalities of steering and visualization. We have integrated the RealityGrid Steering system [36, 37] into LAMMPS in order to introduce these features. The RealityGrid Steering system was designed for such legacy codes to be fully grid enabled. This means that the steering system allows applications to be deployed on a computational grid using the RealityGrid launcher GUI, which then can be steered using a steering client. Further integration of the steering library into a visualizer means that the application can transmit its data to a visualization service. The visualizer itself can be launched on a separate machine to that of the application, and communication is carried out over sockets.

The RealityGrid launcher was built to manage steerable applications in the RealityGrid computational steering framework. To enable a scientist to launch, steer and visualize a simulation on separate grid resources, the launcher has to submit jobs for simulation and visualization, start a variety of supporting services, and put all these loosely coupled components in communication with each other. In doing this it relied on the presence of grid client software (actually Globus commands invoked through customized scripts) on the end-user's machine. This approach possesses several of the drawbacks discussed in this paper, all of which increase the barrier to uptake. These include:

- deep software dependencies make the launcher heavyweight.
- the situation in the client of (customiz-

able) logic to orchestrate the distributed components implicates the end-user in ongoing maintenance of the client's configuration (consider the difficulty of adding a new application or new resource, especially one operating a different middleware, to the set that the user can access).

- the client needs to be "attached" to the grid in order to launch and monitor jobs and retrieve results, which decreases client mobility.

The AHE approach alleviates these difficulties by moving as much of the complexity as possible into the service layer. The AHE decomposes the target audience into expert and end-users, where the expert user installs, configures and maintains the AHE server, and the end-users need simply to download the ready-to-go AHE client. The client itself becomes thinner, and with a reduced set of software dependencies is easier to install. All state persistence occurs at the service layer, which increases client mobility. Architecturally, the AHE is akin to a portal, but one where the client is not constrained to be a Web browser, increasing the flexibility of what the client can do, and permitting programmatic access, which allows power users to construct lightweight workflows through scripting languages.

VII Summary

By narrowing the focus of the AHE middleware to a small set of applications that a group of scientists will typically want to use, the task of launching and managing applications on a grid is greatly simplified. This translates to a smoother end user experience, removing many of the barriers that have previously deterred scientists from getting involved in grid computing. In a production environment we have found the AHE to be a useful way of providing a single interface to disparate grid resources, such as machines hosted on the NGS and TeraGrid.

By representing the execution of an application as a stateful web service, the AHE can easily be built on top of to form systems of arbitrary complexity, beyond its original design. For example, a BPEL engine could be developed to allow users to orchestrate the workflow of applications using the Business Process Execution Language. Employing a graphical BPEL workflow designer would ease the creation of workflows by users not comfortable with creating scripts to call the command line clients, which is something we hope to look at in the future.

In future we also hope to be able to use a GridSAM connector to the Unicore middleware to allow the AHE to submit jobs to the DEISA grid. By providing a uniform interface to these different back end middlewares, the AHE will provide a truly interoperable grid from the user's perspective. We also plan to integrate support for the RealityGrid steering framework into the AHE, so that starting an application which is marked as steerable automatically starts all the necessary steering services, and also to extend the AHE to support multi-part applications such as coupled models. The end-user still deals with a single application, while the complexity of managing multiple constituent jobs is delegated to the service layer.

VIII Acknowledgements

The development of the AHE is funded by the projects "RealityGrid" (GR/R67699) and "Rapid Prototyping of Usable Grid Middleware" (GR/T27488/01), and also by OMII under the Managed Programme RAHWL project. The AHE can be obtained from the RealityGrid website: http://www.realitygrid.org/AHE.

References

[1] P. V. Coveney, editor. *Scientific Grid Computing.* Phil. Trans. R. Soc. A, 2005.

[2] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Intl J. Supercomputer Applications*, 15:3–23, 2001.

[3] http://www.globus.org.

[4] http://www.unicore.org.

[5] J. Chin and P. V. Coveney. Towards tractable toolkits for the grid: a plea for lightweight, useable middleware. Technical report, UK e-Science Technical Report UKeS-2004-01, 2004. http://nesc.ac.uk/technical_papers/UKeS-2004-01.pdf.

[6] J. Kewley, R. Allen, R. Crouchley, D. Grose, T. van Ark, M. Hayes, and Morris. L. GROWL: A lightweight grid services toolkit and applications. 4th UK e-Science All Hands Meeting, 2005.

[7] S. Graham, A. Karmarkar, J Mischkinsky, I. Robinson, and I. Sedukin. Web Services Resource Framework. Technical report, OASIS Technical Report, 2006. http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-os.pdf.

[8] http://www.ccp.ac.uk/.

[9] IETF. *The IP Network Address Translator (NAT).* http://www.faqs.org/rfcs/rfc1631.html.

[10] IETF. *The TLS Protocol Version 1.0.* http://www.faqs.org/rfcs/rfc2246.html.

[11] P. Coveney, J. Vicary, J. Chin, and M. Harvey. Introducing WEDS: a Web services-based environment for distributed simulation. In P. V. Coveney, editor, *Scientific Grid Computing*, volume 363, pages 1807–1816. Phil. Trans. R. Soc. A, 2005.

[12] http://gridsam.sourceforge.net.

[13] http://www.sve.man.ac.uk/research/AtoZ/ILCT.

[14] http://www.soaplite.com.

[15] M. Gudgin and M. Hadley. Web Services Addressing, 2005. http://www.w3c.org/TR/2005/WD-ws-addr-core-20050331.

[16] J. Treadwell and S. Graham. Web Services Resource Properties, 2006. http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-spec-os.pdf.

[17] L. Srinivasan and T. Banks. Web Services Resource Lifetime, 2006. http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-os.pdf.

[18] T. Maguire, D. Snelling, and T. Banks. Web Services Service Group, 2006. http://docs.oasis-open.org/wsrf/wsrf-ws_service_group-1.2-spec-os.pdf.

[19] L. Liu and S. Meder. Web Services Base Faults, 2006. http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf.

[20] M. Gudgin, M. Hadley, N. Mendelsohn, J. Moreau, and H. Frystyk. Soap version 1.2 part 1: Messaging framework. Technical report, W3C, June 2003. http://www.w3.org/TR/soap12-part1.

[21] A. Nadalin, C. Kaler, P. Hallam-Baker, and R. Monzillo. Web Service Security: SOAP Message Security 1.0, 2006. http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf.

[22] J. Brooke, M. Mc Keown, S. Pickles, and S. Zasada. Implementing WS-Security in Perl. 4th UK e-Science All Hands Meeting, 2005.

[23] http://www.cs.wisc.edu/condor.

[24] http://gridengine.sunsource.net.

[25] http://www.omii.ac.uk.

[26] Job Submission Description Language Specification. GGF. http://forge.gridforum.org/projects/jsdl-wg/document/draft-ggf-jsdl-spec/en/21.

[27] http://www.apache.org.

[28] http://tomcat.apache.org.

[29] E. Gamma, R. Helm, R. Johnson, and Vlissides J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[30] IETF. *HTTP Extensions for Distributed Authoring – WEBDAV.* http://www.faqs.org/rfcs/rfc2518.html.

[31] http://www.postgresql.org/.

[32] http://ws.apache.org/axis.

[33] S.J. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. of Comp. Phys.*, 117:1–19, 1995.

[34] L Kale, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulte. NAMD2: Greater scalability for parallel molecular dynamics. *J. Comp. Phys.*, pages 283–312, 1999.

[35] H. C. Greenwell, W. Jones, P. V. Coveney, and S. Stackhouse. On the application of computer simulation techniques to anionic and cationic clays: A materials chemistry perspective. *Journal of Materials Chemistry*, 16(8):706–723, 2006.

[36] S. M. Pickles, R. Haines, R. L. Pinning, and A. R. Porter. Practical Tools for Computational Steering. 4th UK e-Science All Hands Meeting, 2004.

[37] S. M. Pickles, R. Haines, R. L. Pinning, and A. R. Porter. A practical toolkit for computational steering. In P. V. Coveney, editor, *Scientific Grid Computing*, volume 363, pages 1843–1853. Phil. Trans. R. Soc. A, 2005.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Building simple, easy-to-use Grids with Styx Grid Services and SSH

Jon Blower*, Keith Haines

Reading e-Science Centre, Environmental Systems Science Centre, University of Reading, Harry Pitt Building, University of Reading, Whiteknights, Reading RG6 6AL

* Corresponding author: email address jdb@mail.nerc-essc.ac.uk

Abstract

Grid systems have a reputation for being difficult to build and use. We describe how the ease of use of the Styx Grid Services (SGS) software can be combined with the security and trusted nature of the Secure Shell (SSH) to build simple Grid systems that are secure, robust and easy to use and administer. We present a case study of how this method is applied to a science project (GCEP), allowing the scientists to share resources securely and in a manner that does not place unnecessary load on system administrators. Applications on the Grid can be run exactly as if they were local programs from the point of view of the scientists. The resources in the GCEP Grid are diverse, including dedicated clusters, Condor pools and data archives but the system we present here provides a uniform interface to all these resources. The same methods can be used to access Globus resources via GSISSH if required.

1 Introduction

As Grid technology matures, an increasing amount of attention is being devoted to the problem of enhancing the usability of Grid systems. It is a well-known problem [5] that many Grid toolkits (e.g. Globus [6]) are difficult to install and use on both the client and server side. Futhermore, these toolkits often require a large number of incoming ports to be open on the server (and the client), which is sometimes inconsistent with local security policy.

Grid computing is usually defined as distributed computing performed transparently across multiple administrative domains. This definition does not mandate the use of complex Grid middleware. We believe that perfectly functional Grids can be created using simple, well-understood software in a manner that is friendly to both users and resource providers.

1.1 The problem of usability

Despite many advances in Grid usability, there is still much scope for further increasing the ease of use of Grid technology. Users typically have to perform multiple steps of resource selection, file transfer, job submission and job monitoring in order to run even simple Grid jobs. The aim of the Styx Grid Services system [4] is to make it as easy as possible for novice users to use Grid systems by *making the running of applications on a Grid as easy as running applications on the user's own computer*. In this paper we shall describe how the Styx Grid Services system can be adapted for use in more secure environments, whilst retaining its essential quality of being easy to install and use.

1.2 The problem of security

A key problem in the design of Grid systems is security. It is very important to find a level of security with which both users and resource providers can be satisfied. If the security level is too low there will be an unacceptable risk that the resource in question will be compromised by unauthorized access. Conversely, if the security level is too high, many users will find it too difficult to use the resource; in this case, users will either choose not to use the resource in question or, worse, users will find ways to circumvent the

security mechanism.

A good example of this is the Grid Security Infrastructure (GSI [11]), which is based on public-key infrastructure and proxy certificates. This security mechanism gives, theoretically, a high level of security; however, it often raises complaints from users who feel that this level of security interferes with their normal working practices to an unacceptable degree. Many users feel that the processes of caring for their digital certificates, renewing them annually and generating time-limited proxies before running jobs are too cumbersome. This user-unfriendliness leads some users to engage in forbidden practices such as storing their digital certificates in multiple, perhaps insecure, locations and sharing their digital certificates with colleagues [1]. The resource administrator has no control over these practices and cannot detect that they are occurring.

Some Grid systems (e.g. the NERC Data Grid) take the approach of completely insulating the user from the certificate-based security architecture by storing users' certificates in some secure location such as a MyProxy server and allowing the users to log on to the Grid using a username and password. Whenever the certificate is needed, it is automatically retrieved in a manner that is hidden from the user. This "portal" approach is one of the best means for increasing the usability of Grid security but does require a system administrator to care for the certificates on behalf of the users.

1.3 Why not use SSH and SFTP?

At heart, many Grid jobs consist of transferring input files to a Grid resource, running the job and transferring the output files back to the user. A common question that is asked by users is, "Why can't I simply use the Secure Shell (SSH) and Secure FTP (SFTP, or its legacy equivalent SCP) to run my jobs?" SSH and SFTP are well-known utilities with which a large proportion of users are already familiar. Files can be transferred to and from remote resources securely with SFTP. Applications can be run on remote machines through SSH. Importantly, SSH and SFTP are very familiar to system administrators (sysadmins). Sometimes, sysadmins will *only* permit remote access to their resources through SSH.

In order to log on to a remote resource using SSH, the user must have an account on that resource. Therefore, if the user wishes to access many resources, he or she will need an account on each of those resources. At first glance, this would seem to violate the principle of "single sign-on", in which the user can access multiple resources having authenticated only once. Single sign-on is one of the problems that GSI addresses through the use of user certificates. However, in practice many systems that use GSI also require that each user has a unique user account on each resource: the user's certificate is then simply mapped to this account. Single sign-on can be achieved in the SSH domain through the use of public and private keys and authentication agents. Computer-savvy users who dislike the GSI method of authentication have been known to take unusual steps in order to obtain a plain SSH connection (e.g. running an SSH server as a Grid job and then logging into it).

For many Grids, particularly those with a relatively small number of users and for which little administrative effort is available, we believe that the use of SSH in place of more complex systems is perfectly acceptable. A well-administered SSH-based system is likely to be *more* secure than a more complex system that is not properly administered, and in which users engage in poor security practices.

1.4 The proposed solution

In this paper we shall describe how the ease of use of the Styx Grid Services system can be combined with the security of SSH to allow Grids to be constructed simply, allowing users to run Grid jobs with very little effort and placing very little load on system administrators. The combined system is known as SGS-SSH. For resource providers that must use GSI security, GSISSH can be used in place of SSH with little additional effort; no more middleware (e.g. the rest of the Globus Toolkit) is necessary.

We shall illustrate our solution through a case study. The GCEP (Grid for Coupled Ensemble Prediction) project is a NERC e-Science project that will study the predicability of the Earth's climate on seasonal to decadal timescales. In GCEP, users will run jobs on a Grid of machines in several different institutions, including the UK National Grid Service (NGS). Our solution will allow users to run Grid jobs just as easily as if they were running programs on their local machines. Furthermore, the administrators of the machines in the Grid will not be burdened by the tasks of installing and maintaining complex pieces of software.

Before we describe how the SGS-SSH system is constructed, we shall summarize the key

benefits of the Styx Grid Services system for enhancing the ease of use of Grids.

2 How Styx Grid Services make Grids easy to use

The Styx Grid Services (SGS) system is a means for running remote applications *exactly* as if they were installed locally. The software is pure Java, is very small in size (under 3 MB) and is easy to install and configure. The technical details of the SGS system have been discussed in previous papers [3, 4] and will not be repeated here. To illustrate how the SGS system is installed and used, we shall work through a concrete example.

Let us take the example of a simple visualization program called `makepic` that reads an input file and creates a visualization of the results as a PNG file. The names of the input and output files are specified on the command line, for example:

```
makepic -i input.dat -o pic.png
```

This program can be deployed on a server as a Styx Grid Service as follows. The `makepic` program is installed on the server. A simple XML configuration file is created on the server. This file describes the program in terms of its inputs, outputs and command-line arguments (see figure 1). The SGS server daemon is then started.

Clients can now run the `makepic` Styx Grid Service from remote locations, exactly as if the `makepic` program were deployed on their local machines. They do this using the `SGSRun` program, which is a generic client program for running any Styx Grid Service:

```
SGSRun <hostname> <port>  \
   makepic -i input.dat -o pic.png
```

where `<hostname>` and `<port>` are the host name (or IP address) and port of the SGS server respectively. The `SGSRun` program connects to the SGS server and downloads the XML description of the `makepic` program (figure 1). `SGSRun` uses this configuration information to parse the command line arguments that the user has provided. It then knows that `input.dat` is an input file and uploads it automatically from the user's machine to the SGS server before the program is started. Having started the program, `SGSRun` knows that `makepic` will produce an output file called `pic.png`, which it downloads to the user's machine.

It is a very easy task to create a simple shell script (or batch file in Windows) called `makepic` that wraps the `SGSRun` program and contains the host name and port of the SGS server, so the user can simple run:

```
makepic -i input.dat -o pic.png
```

exactly as before. Many scientific codes have corresponding graphical wrapper programs that use the command-line executable as an "engine", display results graphically and perhaps allow the user to interact with the running program. The `makepic` script (which calls the remote Styx Grid Service) behaves identically to the original executable and so could be called by such a graphical wrapper program, thereby "Grid-enabling" the graphical program without changing it at all. More details on this process can be found on the project website [2].

By allowing the user to execute programs on a Grid exactly as if they were local programs, the Styx Grid Services software provides a very natural and familiar environment for users to work in. Users do not have to know the details of where or how the remote program runs, nor do they need to manually upload input files to the correct location: once the program is deployed as a Styx Grid Service, all this is handled automatically.

2.1 Styx Grid Services and workflows

Given that, in the SGS system, remote services can be executed exactly as if they were local programs, simple shell scripts can be used to combine several remote services to create a distributed application or "workflow". This is described in detail in [4]. No special workflow tools are required on either the client or server side to do this. Data can be transferred directly between applications along the shortest network route, saving time and bandwidth (i.e. intermediate data do not have to pass through the client). It should be noted that this method does not provide all the features that might be required of a general workflow system but will be sufficient for many users.

2.2 Running jobs on Condor and Sun Grid Engine via SGS

In the above example the `makepic` program runs on the SGS server itself. By using a distributed resource management (DRM) system such

```
<gridservice name="makepic" command="/path/to/makepic">
  <params>
    <param name="inputfile" paramType="flaggedOption" flag="i" required="yes"/>
    <param name="outputfile" paramType="flaggedOption" flag="o" required="yes"/>
  </params>
  <inputs>
    <input type="fileFromParam" name="inputfile"/>
  </inputs>
  <outputs>
    <output type="fileFromParam" name="outputfile"/>
    <output type="stream" name="stdout"/>
  </outputs>
</gridservice>
```

Figure 1: Portion of the configuration file on a Styx Grid Services server, describing the `makepic` *program that is deployed. This specifies that the program expects one input file, whose name is given by the command-line argument following the "-i" flag. The program outputs one file, whose name is given by the command-line argument following the "-o" flag, and also outputs data on its standard output stream.*

as Condor [10] or Sun Grid Engine (SGE, [8]), the program can be run on a different machine, balancing the load between a set of machines in a Condor pool or cluster. With a very small change to the SGS configuration file, the SGS system can run programs through Condor or SGE in a manner that is completely transparent to the user: the client interface is identical in all cases. Plugins to allow the SGS system to run jobs on other DRMs such as PBS (Portable Batch System) can be created if required.

This is particularly useful when the user wishes to execute the same program many times over a number of input files. This is known as "high-throughput computing" and is commonly used in Monte Carlo simulations and parameter sweep studies. In the above example, the user might wish to execute the `makepic` program over a large number of input files, creating a visualization of each one. Normally this would require the user to upload the input files, name them in some structured fashion and create an appropriate job description file in a format that is understood by Condor, SGE or the DRM system in question.

In the Styx Grid Services system, the running of these high-throughput jobs is very simple. Let us imagine that the user has a set of input files for the `makepic` program in a directory called `inputs` on his or her local machine. The user simply runs the `makepic` Styx Grid Service as before but, instead of specifying a single input file on the command line, the user enters the name of the `inputs` directory:

```
makepic -i inputs -o pic.png
```

where `makepic` is the script that wraps the `SGSRun` executable as described above.

The input files are automatically uploaded to the SGS server as before. The server notices the presence of an input directory where it was expecting a single file. It takes this as a signal to run the `makepic` program over each file in the input directory, producing a picture for each file. The client then downloads these pictures automatically and places them in a directory called `pic.png` on the user's local machine. The SGS server uses the underlying DRM system (e.g. Condor or SGE) to run these tasks in parallel on the worker nodes. The progress of the whole job is displayed to the client as the individual tasks are executed.

2.3 Disadvantages of the SGS system

The design of the SGS system has focused on ease of use, deployment and maintenance. There are a number of disadvantages of the system:

- It only works with command-line programs (this is true for most Grid systems).
- The current SGS system will not work well in cases where the output files that are produced by a program are not predictable, or where the program spawns other programs during execution. Work is underway to address this limitation.

- The SGS system has its own security model. The server administrator must maintain an SGS user database in addition to that of the underlying system.

- On the server, each application runs with the permissions of the owner of the SGS server itself, rather than with the permissions of the specific user.

- Some resource providers might not trust the SGS software without considerable further testing and wider acceptance within the community.

3 SGS-SSH

The last three issues in the above list can be addressed by combining the SGS system with the Secure Shell (SSH). Essentially, the user uses the SGS system exactly as before (and benefits from its ease of use) but mutually authenticates with the server using SSH. All SGS traffic is transmitted over the secure channel that is created by the SSH connection. This combined system is known as SGS-SSH.

This is achieved through a relatively small modification to the SGS software. The original SGS server reads and writes messages through network sockets. The SGS-SSH "server" is actually a program that reads messages on its standard input and writes replies to its standard output. The SGS client is modified to connect to the remote server via SSH and execute the SGS-SSH "server" program via an exec request. The client then exchanges messages with the SGS-SSH server program through the secure channel.

Therefore, the only server that needs to be maintained by the system administrator is the SSH server itself. The user authenticates via the normal SSH procedures and the applications that the user runs through the SGS-SSH interface run with the permissions of the specific user in question, not a generic user. There is no separate user database.

One might ask what the value is of using the Styx Grid Services system at all - why not simply use SFTP and SSH? There are a number of advantages of the SGS system in this environment:

- The SGS system automatically takes care of uploading the input files and downloading the output files to and from the Grid resource.

- The SGS system saves the user from having to manually create job submission files for DRM systems such as Condor and SGE (section 2.2 above).

- The SGS system automatically monitors the progress of jobs and displays this to the user.

- By running the application through an intermediary program (i.e. the SGS server program), there is an opportunity to automatically harvest metadata about each run. One could keep a record of each invocation of a particular program on the Grid and automatically store metadata about each run (this metadata could be provided by the user from the `SGSRun` program).

4 Case Study: the GCEP project

GCEP (Grid for Coupled Ensemble Prediction) is a NERC e-Science project that involves the Reading e-Science Centre (ReSC), the NERC Centre for Global Atmospheric Modelling (CGAM), the British Antarctic Survey (BAS) and the Council for the Central Laboratory of the Research Councils (CCLRC). GCEP aims to test the extent to which the Earth's climate can be predicted on seasonal to decadal timescales. This will be achieved by running a large number of computer-based climate simulations, with each simulation being started from a different set of initial conditions of the oceans and the distributions of ice, soil moisture and snow cover. If any aspect of climate can be predicted on these timescales, it is these slowly-varying properties that will contain the necessary information.

The climate simulation program that will be run is the Hadley Centre's full HadCM3 model [7], which has recently been ported so that it can run on PC clusters. Large numbers (*ensembles*) of simulations will need to be run in order to gather the necessary statistics. The process of analysing the results of the simulations will also be compute- and data-intensive. The compute resources available to the project include clusters at ReSC, BAS and CCLRC, a Condor pool at the University of Reading and the UK National Grid Service (NGS). The project will generate several terabytes of data and so an efficient data management system will be required to allow results to be shared amongst the project scientists.

4.1 GCEP Grid architecture

The GCEP partners wish to focus on the science challenges and do not wish to devote large amounts of time to the installation and maintenance of complex Grid middleware. We propose to use SGS-SSH to build the Grid. In this way, the administrators of the GCEP clusters (at ReSC, BAS and CCLRC) only need to maintain an SSH server and set up a local user account for each of the GCEP users. Figure 2 gives an overview of the proposed GCEP architecture.

There are two types of job that will be run in GCEP. The first type is the running of the HadCM3 model itself to produce simulations of the evolution of the Earth's climate arising from various initial conditions. HadCM3 is a complex model to configure and it only runs efficiently on certain clusters (e.g. it requires a large amount of memory per node). Therefore, we expect that scientists will run the HadCM3 model "manually", by logging into the cluster in question (via SSH) and carefully configuring the program before running it. This is an expert task that is not easily automated.

The second job type that will be run in GCEP is the analysis of the output from the HadCM3 model runs. These data analysis jobs are typically much simpler to set up and can run on a wide variety of machines. It is these data analysis jobs that are most suitable for automation using the Styx Grid Services system. We propose that, when a scientist creates a useful data analysis tool (i.e. a program), he or she arranges for it to be installed on each of the GCEP resources as a Styx Grid Service. All other scientists can then use this tool exactly as if they had it installed locally. The SGS system would allow the tools to be run on a variety of resources (SGE clusters, Condor pools, individual machines) in a manner that is completely transparent to the scientists.

Although this does not use "traditional" Grid software, this is true Grid computing: it is distributed computing, performed transparently across multiple administrative domains.

4.2 Using the National Grid Service

When the GCEP scientists wish to use Globus resources such as the UK National Grid Service they will not be able to use SSH to access these resources directly. There are two approaches to this:

1. Use GSISSH as a direct replacement for SSH: this uses the user's Grid certificate to authenticate with the server. This requires the user to obtain and care for a certificate and create proxy certificates whenever he or she needs to run a job.
2. Create a "gateway" machine, which the user accesses through SSH as before. This gateway machine accesses the Globus resource on the user's behalf, creating the necessary proxy certificate automatically. The user's certificate could be stored on this gateway machine (if it is sufficiently secure) or the certificate could be stored in a MyProxy server and retrieved automatically when the user runs a job.

At the time of writing it is not clear which approach will be more appropriate for the project and further investigation will be required.

4.3 Automatic resource selection

A greater degree of transparency could be obtained by providing a system that automatically finds the best resource in the GCEP Grid on which a given job should run. This might involve automatic selection of the "fastest" or "least busy" machine in the Grid to ensure that the scientists receive their results as quickly as possible. Although an attractive idea in theory, this is very difficult to achieve in practice. We envisage that the GCEP users will, initially at least, have to perform their own resource selection. It may be possible to create a basic resource selector that queries the queue management system on each of the GCEP resources and makes a decision; however, this is unlikely to be optimal as it is extremely difficult to quantify the "load" on any given machine or to estimate the true length of a queue.

4.4 Data management

The GCEP project will produce several terabytes of data, which will be stored at the various partner institutions and which must be shared amongst the project scientists at all the institutions. One option for handling this is to use the Storage Resource Broker (SRB). This would create a single virtual data store that combines all the data stores at the partner locations. Users would not have to know where each individual data file is stored: this is handled by the SRB system. Additionally, the SRB system can store metadata about each file in the

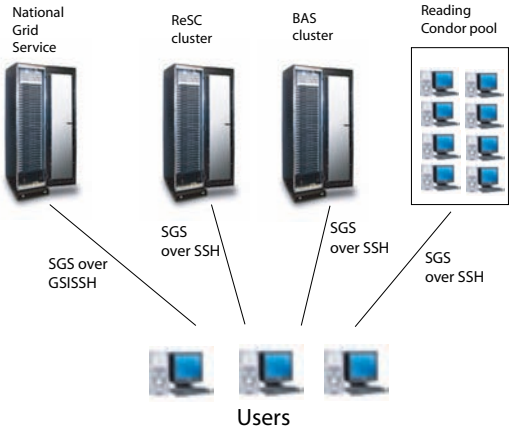


Figure 2: Overview of the proposed architecture for the GCEP project, omitting the CCLRC cluster for clarity. The dedicated GCEP resources simply run SSH servers, as does the Reading Condor pool. The National Grid Service resources are accessed through GSISSH. The resources themselves run a variety of Distributed Resource Management systems including Condor and Sun Grid Engine. The Styx Grid Services software is run over SSH and GSISSH to allow users to run Grid applications on these various resources just as easily as they would run local applications. Files are shared amongst the resources using SSH and sshfs; the Storage Resource Broker is another option for this (section 4.4).

system, making it easier for scientists to find files in the entire virtual store.

An alternative approach, which would sacrifice some functionality but require less installation and configuration, would be simply to share the files via SSH. Thanks to the widespread nature of SSH there are many tools for providing easy access to filesystems on an SSH server. On Linux clients, remote filesystems that are exposed through SSH can be mounted on the client's machine through sshfs [9]. This has some significant advantages over SRB. With sshfs, users would not have to copy entire data files to the location at which a particular data analysis program needs to run: the machine running the program can mount the remote disk via sshfs and operate upon the data file as if it were on a local disk. This means that the data analysis program can extract just those portions of the data file that it needs, saving considerable file transfer time. Sshfs is only available to Linux users (of which there are a large proportion in GCEP), although there is an equivalent, low-cost commercial equivalent (www.sftpdrive.com) available for Windows clients. In the absence of sshfs, users can use SFTP (secure FTP) to move files from place to place. An important limitation of an SSH-based distributed file system is that it lacks a dedicated metadata store.

5 Discussion

We have described how the ease of use of the Styx Grid Services system can be combined with the Secure Shell (SSH) to create Grid systems that are easy to build, use and administer. No complex middleware is required and the barrier to uptake by users is very low: most of the tools involved are already familiar to a large proportion of users. We have discussed how SGS-SSH can be applied to a multi-partner science project (GCEP), allowing compute and data resources to be shared transparently and securely with the minimum of administrative effort. This will provide a reliable and easy-to-use Grid system to the GCEP scientists, allowing them to focus on the considerable science challenges rather than having to spend time learning new tools.

SGS-SSH does not provide every possible feature that might be desirable in a full-scale production Grid. However, we argue that it pro-

vides enough features for most users and the simple nature of the system will, in many cases, more than make up for the missing features. We hope that this will encourage more users and resource providers to create Grid systems to support collaborative science projects.

Acknowledgements

This work was supported by the NERC Reading e-Science Centre grant and the NERC GCEP project grant. The authors would like to thank Bruce Beckles for illuminating discussions on Grid security and the GCEP project partners for helpful suggestions and guidance.

References

[1] Beckles, B., V. Welch, and J. Basney, 2005: Mechanisms for increasing the usability of grid security. *International Journal of Human-Computer Studies*, **63**, 74–101.

[2] Blower, J., 2006: Styx Grid Services. Online, http://www.resc.rdg.ac.uk/jstyx/sgs.

[3] Blower, J., K. Haines, and E. Llewellin: 2005, Data streaming, workflow and firewall-friendly Grid Services with Styx. *Proceedings of the UK e-Science Meeting*, S. Cox and D. Walker, eds., ISBN 1-904425-53-4.

[4] Blower, J., A. Harrison, and K. Haines, 2006: Styx Grid Services: Lightweight, easy-to-use middleware for scientific workflows. *Lecture Notes in Computer Science*, **3993**, 996–1003.

[5] Chin, J. and P. V. Coveney, 2004: Towards tractable toolkits for the Grid: a plea for lightweight, usable middleware. UK e-Science Technical Report UKeS-2004-01, http://www.nesc.ac.uk/technical_papers/UKeS-2004-01.pdf.

[6] Foster, I.: 2005, Globus Toolkit version 4: Software for service-oriented systems. *IFIP International Conference on Network and Parallel Computing*, Springer-Verlag, volume 3779 of *LNCS*, 2–13.

[7] Gordon, C., C. Cooper, C. A. Senior, H. Banks, J. M. Gregory, T. C. Johns, J. F. Mitchell, and R. A. Wood, 2000: The simulation of SST, sea ice extents and ocean heat transports in a version of the Hadley Centre coupled model without flux adjustments. *Climate Dynamics*, **16**, 147–168.

[8] Sun Microsystems, 2006: Sun Grid Engine. Online, http://gridengine.sunsource.net/.

[9] Szeredi, M., 2006: SSH filesystem. Online, http://fuse.sourceforge.net/sshfs.html.

[10] The Condor Project, 2006: Condor. Online, http://www.cs.wisc.edu/condor/.

[11] The Globus Alliance, 2006: Overview of the Grid Security Infrastructure. Online, http://www.globus.org/security/overview.html.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Model Based Visualization of Cardiac Virtual Tissue

J W Handley*, **K W Brodlie***, **R H Clayton**†

* School of Computing, University of Leeds, Leeds LS2 9JT
† University of Sheffield, Department of Computer Science,
Regent Court, 211 Portobello Street, Sheffield S1 4DP

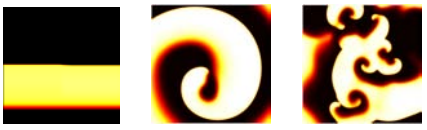
Abstract

In standard analysis of simulations of the heart, usually only one state variable – the trans-membrane voltage (or action potential) – is visualized. While this is the 'most important' variable to visualize, all but the most basic cardiac models have many state variables at each node; data that are not used when visualizing the output. In this paper, we present a novel visualization technique developed within the Integrative Biology project that uses the entire state of the cardiac virtual tissue to produce images based on the deviation from normal propagation of action potential.

1 Introduction

The current generation of high performance computers enable complex simulations of cardiac tissue with anatomically detailed geometries [2, 10] — yet the results of these simulations are usually visualized using a single output from the models: the trans-membrane voltage, or action potential. This leads to images such as those shown in Figure 1 (see also, for example, [8].) In a normal heartbeat the action potential propagates from cell to cell as a plane wave across the tissue (Figure 1 (a)), but this propagation can break down into the circulating pattern of re-entry (Figures 1 (b) and (c)), a potentially lethal malfunction of heart function. Cardiac arrhythmias such as this are an important cause of premature death in the industrialised world, yet the mechanisms that initiate and sustain the lethal arrhythmias of ventricular fibrillation (VF) and ventricular tachycardia (VT) remain poorly understood.

Experimental and clinical studies of VF mechanisms are limited because it is difficult to record electrical activity throughout the 3D ventricular wall, and so most studies are limited to surface recordings — membrane voltage can be imaged on the surface of experimental preparations of heart tissue using voltage sensitive fluorescent dyes. Computational models, however, allow us to examine the whole tissue, and models of action potential propagation in cardiac tissue (cardiac virtual tissues - CVT) have been used extensively in the last decade to probe the mechanisms of VF [7].



(a) (b) (c)

Figure 1: Example visualizations of 2D cardiac virtual tissues with (a) normal propagation, and re-entrant propagation with (b) one and (c) many re-entrant waves visualized using a colourmap where large values are represented by brighter colours.

The action potential is the most important state variable in the models to visualize — the action potential propagates from cell to cell, and acts as a signal for cardiac tissue to contract, i.e. it causes the heart to beat. However, the propagation of the action potential is modulated by the conductance of ion channels in the cell membrane. Slowing or blocking of the action potential can result in re-entry, and so the contribution of these ion channels is important. These cannot be imaged in experimen-

tal preparations, but are available in computational models via state variables. Including these variables in visualizations should provide insight into abnormal cardiac function, such as arrhythmias. In this paper we discuss the issues of visualizing *all* the state variables, and present a technique for doing so.

2 Visualizing Cardiac Virtual Tissue

In the CVT used throughout this paper, action potential propagation is modelled by a reaction diffusion partial differential equation [1]. Several different excitation models can be used, and these range from simplified models with 3 or 4 state variables, to more detailed models with large nonlinear, stiff systems of ODEs and tens of state variables [6]. The equations are solved across a grid, and typical grid geometries are a 2D sheet, a 3D slab, or an anatomically detailed representation of the heart ventricles.

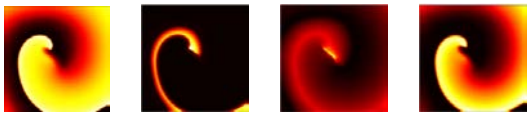


Figure 2: A snapshot of re-entry in a 2D model with excitation described by the 4 variable Fenton Karma model [4]. The four state variables are shown individually using the 'hot' colourmap where low brightness corresponds to low values.

The visualization challenge can be grasped very quickly through looking at the outputs of two different 2D cardiac virtual tissues. Figure 2 shows a snapshot of the state of a 2D CVT in which a re-entrant wave is rotating. In this model excitability is simulated with the simplified 4 variable Fenton Karma model [4] and in this visualization each state variable is visualized separately. Figure 3 shows a similar snapshot of a re-entrant wave in a CVT where excitability is modelled using a modification of the biophysically detailed Luo Rudy

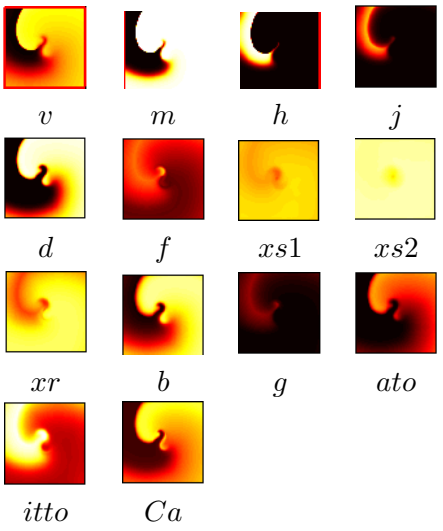


Figure 3: Snapshot of re-entry in a 2D model with excitation described by the biophysically detailed Luo Rudy 2 model [3]. As with Figure 2, each state variable is shown individually, except in this model there are 14 state variables.

2 model [3], which has 14 state variables. It is much more difficult to assimilate the 14 images of Figure 3 into a single mental model of the state of the simulation than the four images of Figure 2.

There are many existing techniques for visualizing multi-variate data, such as parallel coordinates, iconic representations or 'glyphs', and a review of these and other approaches to the visualization of complex data can be found in [9], however none of these techniques successfully handle data that are, and need to remain, four dimensional (i.e. three spatial dimensions and one temporal) while also being (highly) multi–variate. In this paper, therefore, we concentrate on attempting to reduce the data to an uni–variate space, which can then be visualized easily in four dimensions (using animations of isosurfaces, volume rendering, and so on).

Fortunately the dozens of state variables in CVT are not entirely independent, which makes the visualization process easier. This interdependence has two main impacts;

1. Not every state variable will contribute extra information, and some therefore

may be left out of the visualization, and

2. A collection of state variables can be collapsed into a single 'meta-variable' which represents the value of all its constituent variables. For example, the parameters *m*,*h*, and *j* in Figure 3 all depend on membrane voltage and time, and determine the magnitude of current through the Na^+ channel in the cell membrane, which is needed for an action potential to propagate.

The first point can be seen if the correlation co-efficient is calculated between each pair of variables in the four variable Fenton Karma model at each point in time. The correlation used every pixel in a pairwise fashion between each pair of state variables, and can be seen in Figure 4. Note that U and D are almost perfectly correlated across the whole simulation, which suggests there would be little extra information gained through including both of these variables in a visualization.

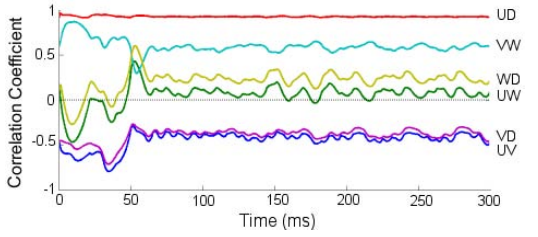


Figure 4: The correlation coefficients of each pair of state variables in the Fenton Karma four variable model across an entire simulation. The two letters at the right-hand end of each trace indicate which pair the correlation refers to.

In this paper, we focus on forming visualizations based on every state variable, with the approaches above highlighted as a future refinement.

3 Visualizing in Phase Space

The nature of the propagation of action potential through the heart imposes a structure of sorts on the data, as each cell goes through a cycle of excitation and recovery. This structure becomes apparent, at least with more simple models, when the data is visualized in phase space. In the case of a tri-variate model, if the standard visualization at time t is to generate three n by m pixel images $U_t(x, y)$, $V_t(x, y)$, and $W_t(x, y)$, for $0 < x < n$ and $0 < y < m$, the phase space visualization is a single 3D scatter-plot image obtained by plotting $n \times m$ 'dots', one at each $(U_t(x, y), V_t(x, y), W_t(x, y))$. Figure 5 shows examples of phase space plots from a Fenton Karma three variable model exhibiting normal propagation.

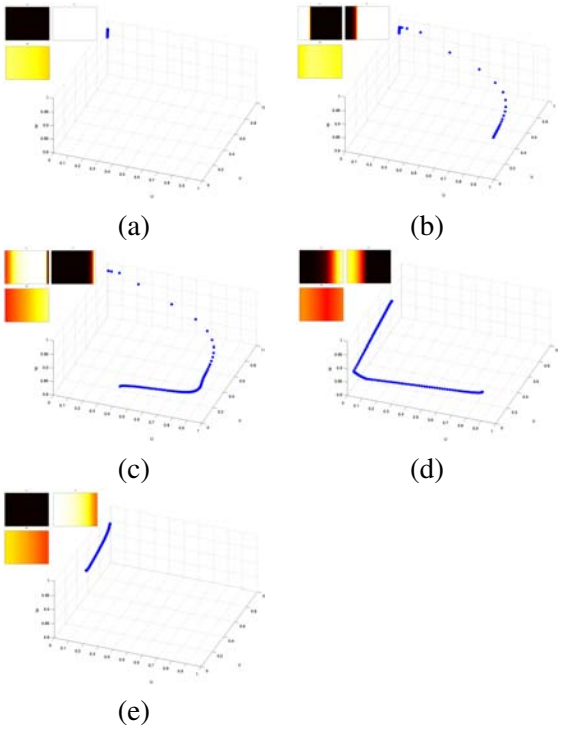
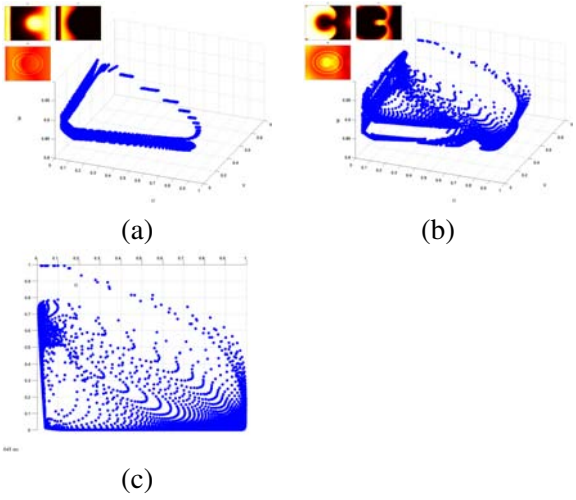


Figure 5: A phase space plot of the 3 parameters from a 2D Fenton Karma three variable model, showing the normal propagation of a wave of action potential at various intervals. The insets show the false colour images of the three variables. (a) rest state – the points occupy $[U = 0, V = 1, W = 1]$, (b) 30 ms after wave initiation along the left edge of the medium, (c) 90 ms, (d) 150 ms, (e) 210 ms.

In Figure 5 the inherent structure is clearly visible as a 'snake' traversing a circuit in phase

space. While this 'snake' is entirely expected – it occurs as cells depolarise then recover – it nevertheless provides an interesting view on wavefront propagation, and should be present even in higher variate models.



(a) (b)

(c)

Figure 6: A phase space plot of the 3 parameters from a 2D Fenton Karma three variable model, showing re-entrant behaviour. (a) The initiation of the second stimulus — the stimulus that causes re-entry. (b) 145 ms later, when re-entry is well established. (c) A plan view of (b).

Perhaps more interesting effects are observed when re-entry is induced in the model. As Figures 6 (b) and (c) shows, the points that previously followed a nominal circumference have now collapsed to almost entirely fill the enclosed space.

Even this simple phase space 'snake' provides a novel view of the structure of the model, particularly when re-entrant behaviour is starting or stopping, and it should also be useful with higher-variate models through judicious choice of the subset of variables plotted. It should be noted that adjacent points in real space can be joined in phase space, but we found this cluttered the images without providing any further insight — an intuitive 'join–the–dots' interpretation being the correct one.

Clearly the phase space is trivial to visualize with two or three variables, but in the case of the Luo Rudy 2 model, with 14 variables, this approach is non-trivial. If the dimensionality reduction techniques from section 2 reduced the phase space even by a factor of three, there would still be too many variables to visualize directly. One approach is to create multiple phase space plots, which has the advantage that all the data are shown, but the disadvantages of still having to visually combine many data sources (see Figure 8, for example). The phase space plots also have the disadvantage of removing spatial information from the visualization.

One way forward is to form visualizations based on the density and/or location of the points in the phase space, through 'hyper-histogram's [5], but we found this to be a less promising approach than using the propagation-model described in this paper.

4 Propagation-Model Based Visualization

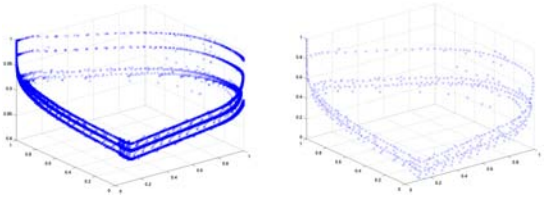
The phase space 'snake' provides an expected behaviour of normal propagation, which in turn suggests the possibility of measuring the deviation from this behaviour. A number of metrics are possible for measuring this deviation — in this study we normalised the output from the simulation so that the range of each state variable was $[0-1]$, and then measured deviation as the Euclidean distance to the nearest point in the model. Ideally, the deviation would be measured as the distance from where the simulation point *ought* to be, but once re-entry is established the concept of the expected state for any given cell becomes meaningless. The nearest 'normal' point acts rather as an approximation to where the point would be if all were normal.

The model of normal propagation was built by capturing every point in n-dimensional phase space for a heart model simulation displaying normal action potential propagation from one edge of the domain to the opposite, for three stimuli. This model was then decimated in phase space to reduce the size of the model from several million points to a few hundred points. This decimation was motivated by two factors; firstly a large number of the points in the model were co-located

in phase-space, at least in part due to the deterministic and quantised simulations being used. Secondly, the computational overhead of performing several hundred thousand distance calculations for every node of the simulation at every point in time would be prohibitive.

The decimation was carried about by iteratively combining all model points within a radius ρ of one another into a single point at the mean location of all those points. ρ was chosen empirically to be the largest value that visually captured, in phase space, the essence of the model. Note that in this application it is not desirable to decimate by phase-space density, as the model is trying to capture a path through space, not a relative expectation of position in phase-space.

Figure 7 shows the complete and decimated models of normal propagation for the Fenton Karma three variable simulation. ρ was 0.02, which reduced the model from 5.4 million points to 528 points. Clearly the decimated model fails to capture the entire region of 'valid' phase space; but the essence is captured, and we found the results with this model to be insightful, as described in Section 5.1.



(a) (b)

Figure 7: The model of normal propagation for a Fenton Karma three variable simulation. (a) The full model, and (b) the decimated model capturing the essence of figure (a).

The same process was used for the Luo Rudy 2 simulation, except that a radius of 0.01 was used, resulting in a model of 831 points. The model is partially shown in Figure 8, which displays the model by projecting it onto each of 13 axes formed by the action potential and every other state variable. Note that this figure offers some support for the existence of the phase space 'snake' in higher variate models.

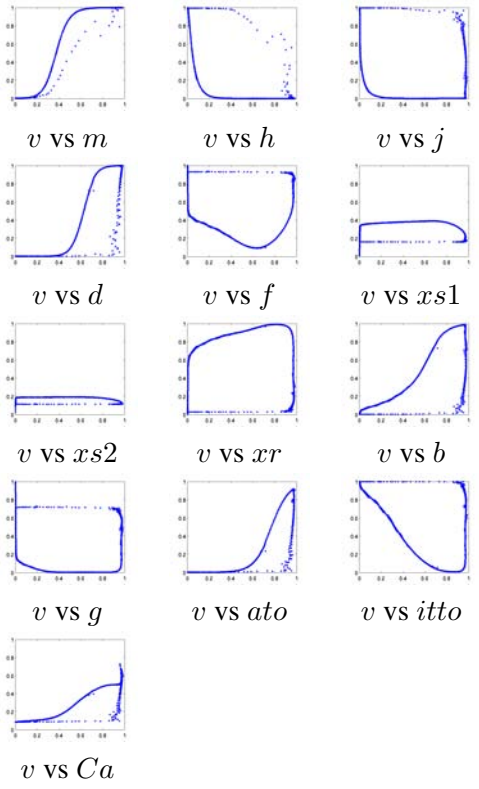


Figure 8: The model of normal propagation for a Luo Rudy 2 fourteen variable simulation. Only a small subset of the model is shown, by plotting the action potential pairwise with every other state variable. The action potential (v) is on the x-axis in every subplot.

Each data point of CVT is assigned a value based on its Euclidean distance from the nearest point of the propagation model. The resulting scalar values, in the range $[0 - \sqrt{n}]$ for an n-dimensional model, can be used, at least in this 2-D case to generate false colour images using a standard colourmap.

5 Results and Discussion

As expected, the visualizations of normal propagations were nearly uniform images of black, as all the distances were around zero. This can be seen in Figure 9. The small deviations from zero were due to the decimation process introducing errors.

Figure 9: The visualization of the deviation from the model of normal propagation for a normal propagation in the Fenton Karma three variable simulation. (a) The action potential output, and (b) the deviation from the model.

As re-entry was introduced in the models, the deviations from the model grew larger, and more interesting images resulted.

5.1 Fenton Karma three variable

Re-entrant behaviour was introduced in the Fenton Karma three variable simulation by changing the tissue type in the central region such that the recovery following an excitation was delayed. In this way spiral waves are initiated when two stimuli are applied in close succession. The onset of re-entry is shown in the sequence of images shown in Figure 10, which are images of the action potential and the deviation from the model at 10 ms intervals,

The first feature to note in Figure 10 is that, even when re-entry is well established, nearly all the tissue in the simulation is operating within the expected model for normal propagation, that is it appears dark. The tissue that is deviating from the normal model is very clear at the leading edge of waves of propagation. This is the tissue that has not yet fully recovered from an early excitation, and is therefore inhibiting or slowing the propagation of the current excitation. This suggests the combining of the three state variables into a single image is providing an insight into the entire state of the model, in that it becomes possible to predict how the action-potential will propagate in the immediate future — something that is not easily assessed from the action-potential images alone. The second feature that can be seen on close inspection of Figure 10 is that the circular heterogeneity in the centre of the domain has become visible in this visualization, whereas it can not be seen in action potential images alone.

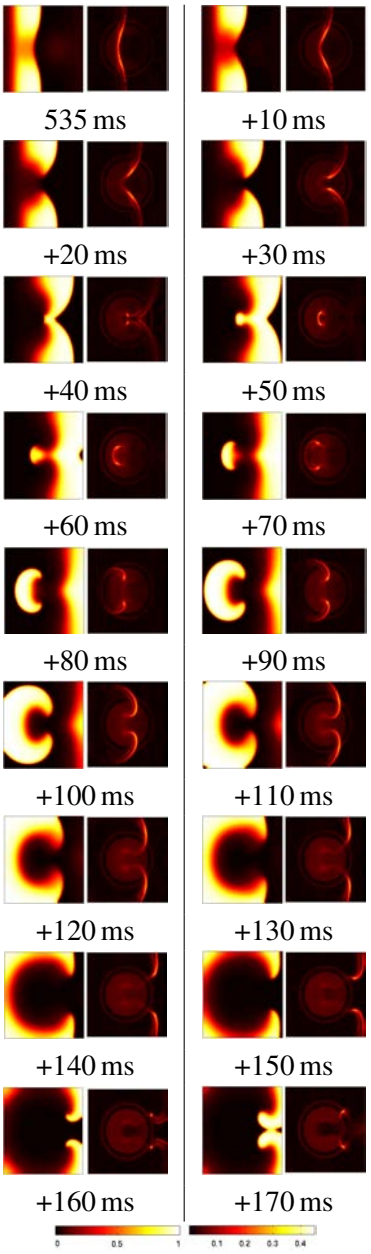


Figure 10: The visualization of the deviation from the model of normal propagation for re-entrant behaviour in the Fenton Karma three variable simulation. The pairs of images show the initiation of re-entry following an early stimulus. The left hand image of each pair is the action potential, and the right-hand the deviation from the model. The colour-bars, shown at the bottom, correspond to the action potential images (left), and the deviation images (right).

5.2 Luo Rudy 2 fourteen variable

Re-entrant behaviour was introduced in a different way for the Luo Rudy 2 simulation — instead of using heterogeneous tissue, the simulation was started in an initial condition where a re-entrant spiral wave immediately forms and continues. The formation of the wave from this artificial start condition was not of interest, so the visualizations were only generated after a settling down period. Figure 11 shows snapshots at 10 ms intervals of a single revolution of the re-entrant wave, with the action potential and the deviation from the model being displayed as before.

The features of this visualization observed in the previous section are present here, in that the re-entrant wavefront shows the largest deviation from normal, and in particular the tip of the spiral wave. This again represents the regions where propagation is being blocked or delayed. It is interesting to note the very slow drop off of this deviation behind the wavefront – behaviour which contrasts with the very sharp drop-off on the Fenton Karma simulation. This may be due a failure of the propagation model to capture normal behaviour, or or that this form of re-entry genuinely does differ to this extent from normal propagation. In any case, the deviation from the model seems to be far less specific than for the Fenton Karma model.

In these visualizations, the deviation from the model gives a very good impression of the excitation state of any given region of tissue; so much so the action-potential images are not really required in this figure.

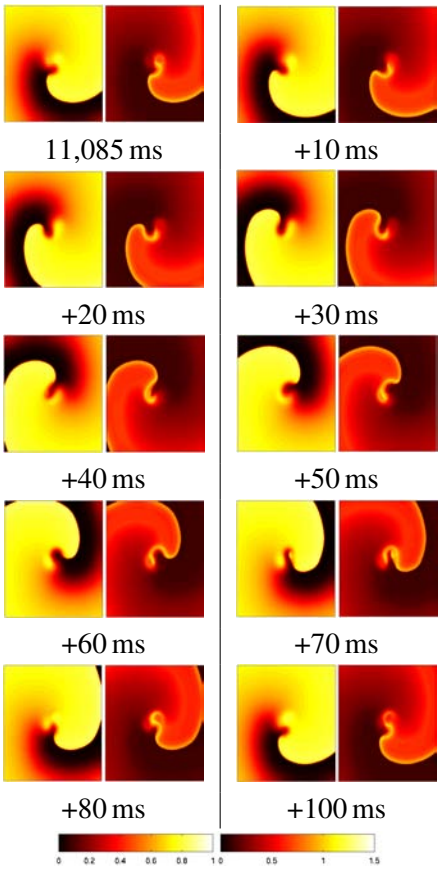


Figure 11: The visualization of the deviation from the model of normal propagation for re-entrant behaviour in the Luo Rudy 2 simulation. The images show a complete revolution of the single spiral wave, with the action potential on the left, and the deviation from the model on the right. The colour-bars are shown at the bottom, with the left colour-bar corresponding to the action potential images, and the right colour-bar to the deviation images.

6 Conclusions and Future Work

This paper has presented a novel method of visualizing cardiac virtual tissues, through the generation of a model of normal propagation in phase space, and measuring the deviation from this model. In the resulting visualizations, regions of CVT where the propagation of action potential is being delayed are highlighted. When combined with the visualization of action potential, this provides insight into how the action potential will propagate through the tissue.

The method for calculating the deviation from the model has scope for further investigation. For instance, it might be possible to calculate the trajectory of a point in phase space during re-entry, and find the nearest model point along this trajectory (either forwards or backwards). There is also further work required on issue of normalisation — for instance is a simple intra-variate normalisation

technique appropriate? There may also be scope in normalising the deviation to be in the range $[0-1]$ rather than $[0-\sqrt{n}]$ so that the results from different models can be directly compared.

This visualization technique can be extended to three-dimensional simulations fairly easily. The distance metric is dimension independent, as it is based in phase space. The significant distances occur at the wave-front of action potential, so an interesting visualization might be to form an iso-surface of action potential, and colour it according to the deviation from the model.

Acknowledgements

The authors wish to acknowledge the support provided by the funders of the UK e-Science Integrative Biology Project: The EPSRC (ref no: GR/S72023/01) and IBM.

References

[1] R.H. Clayton. Computational models of normal and abnormal action potential propagation in cardiac tissue: Linking experimental and clinical cardiology. *Phys. Meas.*, 22:R15–R34, 2001.

[2] R.H. Clayton and A.V. Holden. Filament behaviour in a computational model of ventricular fibrillation in the canine heart. *IEEE Trans. Biomed Eng.*, 51:28–34, 2004.

[3] R.H. Clayton and A.V. Holden. Propagation of normal beats and re-entry in a computational model of ventricular cardiac tissue with regional differences in action potential shape and duration. *Progress in Biophysics and Molecular Biology*, 85:473–499, 2004.

[4] F.H. Fenton, E.M. Cherry, H.M. Hastings, and S.J. Evans. Multiple mechanisms of spiral wave breakup in a model of cardiac electrical activity. *Chaos*, 12:852–892, 2002.

[5] J.W. Handley, K.W. Brodlie, and R.H. Clayton. Multi-variate visualization of cardiac virtual tissue. In *Proc. 19th IEEE International Symposium on Computer Based Medical Systems*, 665–670, IEEE Computer Society, 2006.

[6] D. Noble and Y. Rudy. Models of cardiac ventricular action potentials: Iterative interactions between experiment and simulation. *Philos. Trans. R. Soc. Lond. Ser. A-Math. Phys. Eng. Sci.*, 359:1127–1142, 2001.

[7] A.V. Panfilov and A.M. Pertsov. Ventricular fibrillation: Evolution of the multiple-wavelet hypothesis. *Philos. Trans. R. Soc. Lond. Ser. A-Math. Phys. Eng. Sci.*, 359:1315–1325, 2001.

[8] Blanca Rodríguez, Brock M. Tice, James C. Eason, Felipe Aguel, Jr. Jos M. Ferrero, and Natalia Trayanova. Effect of acute global ischemia on the upper limit of vulnerability: a simulation study. *Am. J. Physiol. Heart. Circ. Physiol.*, 286(6):H2078–H2088, 2004.

[9] P. Wong and R. Bergeron. 30 years of multidimensional multivariate visualization. In Gregory M. Nielson, Hans Hagan, and Heinrich Muller, editors, *Scientific Visualization - Overviews, Methodologies and Techniques.*, pages 3–33, Los Alamitos, CA., 1997. IEEE Computer Society Press.

[10] Fagen Xie, Zhilin Qu, Junzhong Yang, Ali Baher, James N. Weiss, and Alan Garfinkel. A simulation study of the effects of cardiac anatomy in ventricular fibrillation. *Journal of Clinical Investigation*, 113:686–693, 2004.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Service-Oriented Approach to Collaborative Visualization

Haoxiang Wang, Ken Brodlie, James Handley, Jason Wood

School of Computing, University of Leeds, Leeds, LS2 9JT, UK

Abstract

This paper presents a new service-oriented approach to the design and implementation of visualization systems in a Grid computing environment. The approach evolves the traditional dataflow visualization system, based on processes communicating via shared memory or sockets, into an environment in which visualization Web services can be linked in a pipeline using the subscription and notification services available in Globus Toolkit 4. A specific aim of our design is to support collaborative visualization, allowing a geographically distributed research team to work collaboratively on visual analysis of data. A key feature of the system is the use of a formal description of the visualization pipeline, using the skML language first developed in the gViz e-science project. This description is shared by all collaborators in a session. In co-operation with the e-Viz project, we generate user interfaces for the visualization services automatically from the skML description. The new system is called NoCoV (**No**tification-service-based **Co**llaborative **V**isualization). A simple prototype has been built and is used to illustrate the concepts.

1. Introduction

Visualization is widely recognized as a critical component in e-science, allowing insight into the increasingly large datasets generated by simulation and measurement. In recent years a number of important visualization tools have been developed, many of these following the dataflow paradigm. This dataflow approach sees visualization as a sequence of processing steps, whereby raw data is first filtered in some way, then transformed to a geometric representation and finally this geometry rendered as an image. Visualization software provides these steps either as classes that can be embedded in a user program (vtk – vtk, 2006 - is an example of this approach), or as an overall environment with a visual programming editor that enables pipelines to be built from a supplied set of modules (here IRIS Explorer – IRIS Explorer, 2006; Walton, 2004 - is an example). These pipelines can be built, torn apart and reformed, as users experiment with different ways of looking at their data. The dataflow paradigm for visualization (first suggested 20 years ago) has stood the test of time, not least because it provides a high level of abstraction for designing visualization applications.

Major computational applications today typically involve distributed computing – with the user interface executed at the desktop, and remote resources used for computationally demanding tasks. Some traditional visualization systems have managed to evolve to this style of working: in fact IRIS Explorer was designed from the outset to have this capability, with the visual editor on the desktop controlling remote execution of modules. Recent work in the gViz e-Science project (gViz, 2006) has exploited this to adapt IRIS Explorer to grid computing.

However there is a trend today to employ Service-Oriented architectures in designing distributed computing applications. An important and pioneering effort to utilize web services in visualization was the GAPtk toolkit (Sastry and Craig, 2003) which provides specific 'turnkey' visualization services such as isosurfacing, but without the ability to chain them together in pipelines. However, the traditional dataflow visualization paradigm is an excellent match to the concept of a service-oriented architecture: the modules simply become services. Charters (Charters et al, 2004; Charters, 2006) has described the design of a visualization system based on these concepts, in particular using Web Services. Our work in this paper takes this approach further, by using stateful Web services and the facilities in Globus Toolkit 4 (GT4, 2006) for subscription and notification.

Much e-science is multi-disciplinary in nature, involving geographically distributed research teams, and so visualization tools must be designed to be used collaboratively. Again the dataflow paradigm has proved extremely flexible, and it has been exploited in a number of different ways to provide collaborative visualization systems. This work however pre-dates the emergence of Service-Oriented

architectures, and so it is important to study how best to provide team-working in this newer context. We see collaboration as fundamental, and so our design incorporates multi-user working from the outset. The style of collaborative working has evolved from our experiences with collaborative visualization over the past decade.

The structure of the paper is as follows. We begin in section 2 with the vision which underpins our research programme in service-oriented visualization, followed in section 3 by an overview of NoCoV, our proposed system. The design of the system is discussed in section 4, and section 5 describes an implementation of a prototype, developed as proof of concept. Conclusions and future work are in section 6.

2. Service-oriented Visualization – The Vision

The overall vision for our design is a next generation visualization system – based on the proven concept of dataflow pipelines linking elementary visualization modules, but exploiting modern ideas from service-oriented architectures, and involving collaboration between users and between providers, at a global level. Thus we see visualizations being created from a worldwide repository of visualization services, being assembled collaboratively by research teams, and exploiting the latest Grid computing technologies.

A fundamental concept therefore in our design is the Visualization Web Service. This corresponds to a module in a traditional Modular Visualization Environment, or MVE, such as IRIS Explorer or Open DX. In an MVE, modules can be linked in a dataflow pipeline, this being achieved typically by a visual programming 'front-end'. Often, this front-end also provides a user interface to each module, allowing parameters to be modified interactively. We retain the front-end concept, but make a clear distinction between the editing of pipelines and the user interface to services – we envisage visualization application developers having access to pipeline editing, while visualization users only require access to parameter setting, in pipelines previously created.

A central aspect of our vision is a formal description of the visualization pipeline; this indicates the services, the user specified parameters of the services, and the linkage between services. This description, together with information about the expertise of the user, is used to build automatically tailored user interfaces.

The flow of data between modules in an MVE is handled either by shared memory (on the same machine) or by socket connection (between machines). The triggering of dataflow is handled by a firing algorithm. We are able to exploit a novel concept in web services to initiate dataflow, namely notification. A service subscribes to a data element on another service, and receives a notification when that element changes – thus data can flow from one service to another.

Visualization expertise is distributed across the world, and so our vision is to see a worldwide repository of services, maintained on a co-operative basis by specialist groups. These can be wrapped as Grid Archives (gars), listed in a central UDDI, and downloaded and deployed on a local basis by visualization service providers. Thus a flow visualization service developed by a team in the Netherlands could be deployed in Japan by a Japanese service provider.

In any pipeline, each service can be deployed on a different resource – thus allowing us to exploit dedicated rendering resources for example. In general, delivery to the desktop should allow a choice of remote rendering (delivery of images) or local rendering (delivery of geometry).

There is increasing interest in collaborative applications and so a fundamental design requirement of our system is to support team working, where the members of the team may be distributed in space and time. A number of approaches to collaborative visualization using traditional MVEs have been suggested. These include (at two ends of the spectrum): VNC (RealVNC, 2006), where the display of the MVE of one user is shared by a number of collaborators – this is fairly effective when the collaboration is passive, but is extremely awkward if several people wish to take an active role; and COVISA (Wood et al, 1997) where each user develops their own pipeline but can tap data from any point in their pipeline and make it available to all collaborators – this is extremely flexible, and almost any style of collaboration can be programmed, but the different pipelines make it difficult for any user to have a global view of the set of individual pipelines. Thus we aim for a midway position: we have a shared ability to build the pipeline, but there is a single pipeline for all users.

In addition to this *synchronous* (same time, different place) collaboration, there is a similar need to support *asynchronous* working, where

the participation is spread over a period of time. The requirement here is for a persistent pipeline of services, which a collaborator can pick up and develop at a later time – important for collaboration with Australasia for example.

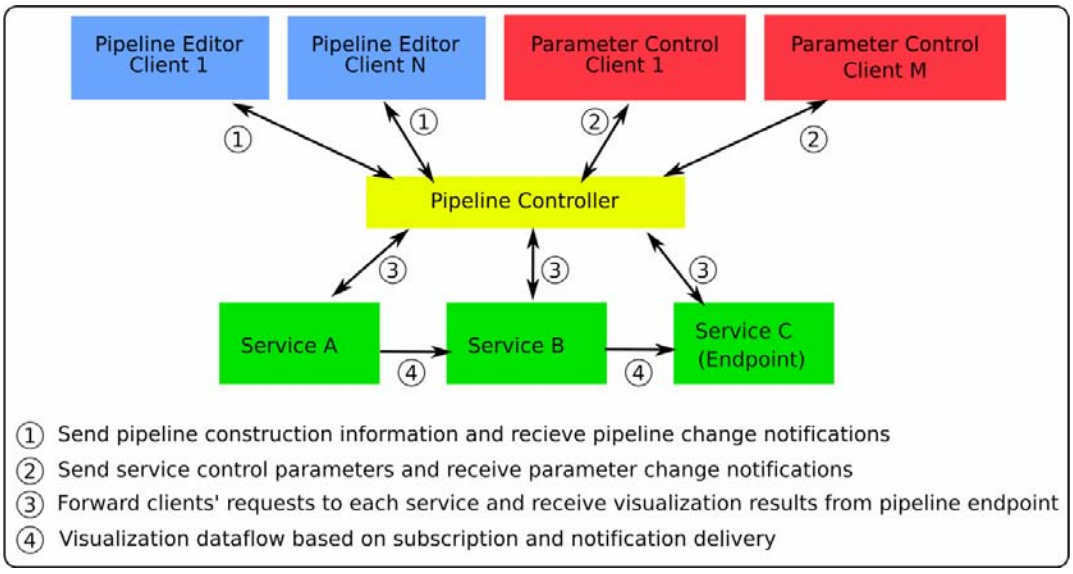


Figure 1: The overview of service-oriented collaborative system (NoCoV)

3. NoCoV System overview

NoCoV (Notification-service-based Collaborative Visualization) provides a collaborative visualization system implemented using Notification Web Services. It is an evolution of the MVE, implemented using a Service-Oriented architecture. Here, visualization services replace the visualization modules of the MVE, and the links between traditional visualization modules are implemented as subscriptions and notifications between visualization services.

As shown in Figure 1, the Pipeline Editor Client is the user interface for creating a visualization pipeline, and the Parameter Control Client provides the GUI for users to interact with parameters on each service. This separation of interfaces allows us to support various classes of users. Visualization experts may use the Pipeline Editor to create visualization applications for use by novice users who subsequently only interact with these pipelines through the Parameter Control Client. A middle class of users may be comfortable creating their own pipelines so may use both interfaces.

The Pipeline Controller Service sits between the end-user clients and the distributed visualization services thus making the visualization services transparent to end-users. It forwards the requests from clients to corresponding services and broadcasts visualization results to subscribing clients as notifications. Users only need to consider the visualization process at a logical level without being aware of the physical locations of these services or the different invocation methods required. The Pipeline Controller Service also acts as a collaborative workspace by sharing pipeline and control parameters with subscribing clients.

The visualization dataflow is implemented by making subscriptions between different visualization services (service A, B and C in Figure 1). Each visualization service publishes one or more notification topics which act as the output ports for that service through which data is sent to other connected services. There is a special service set as an endpoint within each pipeline to which the Pipeline Controller subscribes. This service triggers a notification every time a new result is generated by the pipeline.

NoCoV is an extendable visualization system since customised visualization services can be introduced into the system, so long as the Pipeline Controller service knows how to communicate with these services (i.e. it is provided with the WSDL descriptions of the services). The Pipeline Editor Client lists these customised services as available services for users to link into their pipeline (i.e. services register on a UDDI server from which the Pipeline Editor Client can retrieve the available service list).

To support collaborative working over the construction of a pipeline there is a requirement to share pipeline description information between the Pipeline Controller Service and all the clients. An extension of skML (Duce and Sagar, 2005), an XML based visualization pipeline description language, is used for this purpose. It has been extended to fit the NoCoV system with an emphasis on Service-Oriented features.

The NoCoV system has been implemented with GlobusToolkit 4 (GT4). The stateful Web Services provided by GT4 offer the capability of maintaining visualization pipeline information between sessions. This allows users to save the current status of the pipeline on the Pipeline Controller Service for use the next time they reconnect. We also exploit Notification Web Services in GT4. Moreover, GT4 provides a set of Grid security specifications which can be seamlessly applied to the NoCoV system in its future development to address one of the desired issues in collaboration: security.

4. NoCoV System design

4.1 Using Notification Web Service

WS-Notification includes a set of specifications (OASIS, 2006) and a whitepaper (Publish/Subscribe, 2006) which describe the use of a topic-based publish/subscribe pattern to achieve notification with standard Web Services.

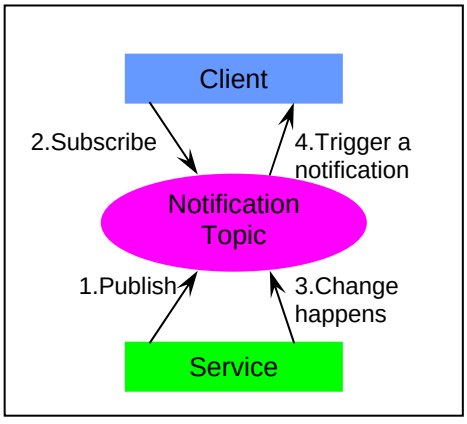


Figure 2:The working pattern of Notification Web Service

With the normal request/response communication mode between service and client, the client has to keep polling the service in order to get the latest changes. By contrast, the notification approach works in the manner shown in Figure 2. When the service publishes a set of notification topics, clients can subscribe to relevant topics according to their different interests. Every time a change happens in these notification topics, the service will automatically deliver the changed information to the subscribing clients.

The main reason for choosing Notification Web Services to implement visualization is their 'push' feature. It fits well with the requirement of a visualization pipeline which needs the services to send their results to other services connected to them 'downstream', each time they produce a new result.

The publish/subscribe pattern provides a data sharing approach for collaboration. The notification topics published by the visualization service can be either the data or the control parameters. Actions from participants (such as changing parameter values) are also published as notifications, allowing these to be shared.

The stateful feature inherited from GT4 Web Services makes it possible to achieve asynchronous collaboration as the status of the pipeline is persistent and users can retrieve the saved pipeline to carry on their previous work – or new users can take over and continue the development.

The publish/subscribe pattern can also reduce network traffic by only delivering changed information to clients. Moreover it can cut down service and client workloads as clients only need to subscribe to the service once and then just wait for the notification messages. It is similar to previously used technologies such as socket communication, but with the facility of WS-Notification, the system developers do not need to consider the details of physical communication ports, and the communication is relatively easy to set up compared to socket communication.

As XML/SOAP messages have a restriction on their maximum size, when large data (e.g. update of geometry) needs to be sent as a notification, only a reference to the data is included in the notification message, and the data itself can be transferred using http, ftp or gridftp separately. Another possible solution is to add data as an attachment (e.g. DIME) to the notification message.

There are however some disadvantages of notification services. When a client subscribes to a notification service, the client needs to be able to function as a listening service waiting for the notification. In the case of a GT4 implementation, it requires GT4 to be installed on the machine where the notification client runs. In addition, every time a change happens on a notification service, the service needs to start a connection to the subscriber. If the

notification client (subscriber) sits behind a firewall, the firewall may block all the connections initiated from outside. In this case, although the client can subscribe successfully to the service, it can not receive any notifications from outside of the firewall. WS-Messenger (Huang, et al, 2006) proposed an approach to address the issue of the delivery of notification through a firewall by using a MessageBox service to store all notification messages and having consumers periodically pull notification from the message box. Another alternative, which is less secure but more straightforward, is to set a small range of open ports in the firewall and configure the local notification system to only use ports within this range.

4.2 The Extended skML

As the proposed NoCoV system is expected to enable the collaborative creation and configuration of visualization pipelines, and the persistence of pipeline information, the visualization pipeline must be represented in such a way that it can be stored as service resource properties.

skML is an XML-based dataflow description language (Duce and Sagar, 2005) which describes visualization pipelines in a generic way, so that the skML description can be independent of the implementation of the pipeline. The skML language was developed as part of the gViz e-science project, and was heavily influenced by MVEs such as IRIS Explorer. However it lacks certain features to describe characteristics of Service-Oriented collaborative visualization pipelines. Rather than create a different description language, we have chosen simply to modify and extend skML.

An ‘instance’ element replaces the ‘module’ element in the skML to represent visualization service instances, but the ‘link’ element in skML is kept to represent the subscriptions to notifications. One of the significant differences is that the extension aims to present richer information about the visualization pipeline. For example, all the output and input ports for each service are recorded: this is then made visible at the user interface, to allow users the ability to change the connections to/from that service. In contrast with the original skML, all control parameters for a service instance must be explicit in the description, again so that they can be presented in an automatically generated user interface. Another difference is the adding of new properties such as ‘owner’ and ‘sharable’, which identify who owns this visualization service instance and who are allowed to access this service instance. These new properties will make it possible to add security control in NoCoV.

4.3 Pipeline Controller Service

A Pipeline Controller Service is placed between the end-users and the visualization notification services, in order to enable users to collaboratively build the pipeline and configure each visualization service linked within the pipeline. Figure 3 displays the components of the Pipeline Controller Service.

The Pipeline Controller stands between the distributed visualization services and the end-users as a proxy, through which users can send requests to create/destroy, connect/disconnect service instances and set parameters of these instances. The removing or adding of visualization services or any changes inside visualization services are transparent to end-users.

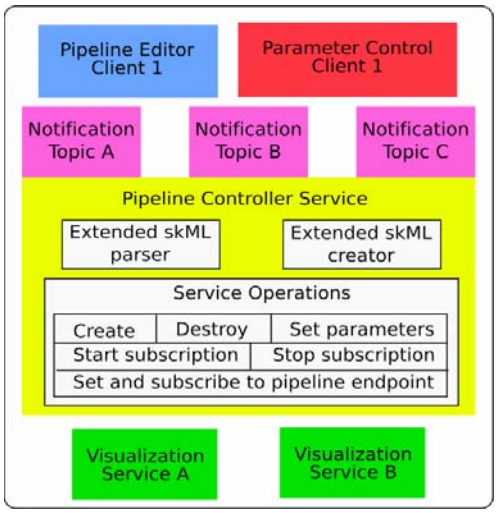


Figure 3: The Pipeline Controller Service

The Pipeline Controller also functions as a shared workspace for all the participants. For visualization experts who create visualization pipelines, the Pipeline Controller keeps a description of the current pipeline in the extended skML, which can be retrieved for the later joiners to the collaborative session. The Pipeline Controller is implemented as a notification service, in which a notification topic about the latest status of the jointly created pipeline is published so that users can share the same pipeline and join in synchronous collaboration of the construction of a pipeline.

Notification topics about control parameters and the latest visualization result generated from the pipeline, are published for scientists who do not want to be involved with the building of the pipeline but simply wish to modify control

parameters. By subscribing to these notification topics, scientists can jointly control the distributed visualization services and view the resulting geometry (in case of local rendering) or image (in case of remote rendering).

4.4 Collaborative Visualization Clients

As mentioned in the previous section, the end-user interfaces are separated into a Pipeline Editor Client and a Parameter Control Client, which only provides users with a parameter control interface rather than the view of whole pipeline.

The Pipeline Editor Client allows users to collaboratively select suitable visualization services and link them in an appropriate way. The Parameter Control Client initializes its GUI from the pipeline description retrieved from the Pipeline Controller, presenting a separate tab corresponding to each service instance created in the pipeline. Users can view parameters on the services and steer the service by changing the parameters through the Parameter Control Client. Code from the e-Viz project (Riding et al, 2005) is used to generate the GUI from the extended skML pipeline description – see section 5.4 for more information.

5. Prototype Implementation

A prototype was implemented as a proof of concept for the NoCoV system. The prototype involves a set of simplified visualization services, a Pipeline Controller Service with basic functions as proposed in the design, a Pipeline Editor Client for visualization experts and a Parameter Control Client for scientists.

5.1 Visualization Services implemented as Notification Web Services

In order to demonstrate the capability of building and controlling a visualization pipeline through end-user interfaces, four visualization services were created with simple visualization functions.

The *data service* retrieves data from a source according to the file name or URL provided by the user. The output of the data service can be subscribed to by an *isosurface service* or a *slice service*, which can generate output geometries in VRML format. The *inline service* can subscribe to both isosurface and slice services to combine multiple geometries into a single scene.

5.2 Pipeline Controller Service

The Pipeline Controller acts as an agent for end-users, releasing users from the burden of dealing directly with visualization services. In the prototype, the Pipeline Controller service can create instances from visualization services, connect instances to build up a visualization pipeline and subscribe to the endpoint of visualization pipeline to receive newly generated visualization results.

The Pipeline Controller has the following notification topics: a representation of the current pipeline information including which visualization instances have been created, how these instances are connected to each other and the setting of control parameters for each instance; the latest result generated from the pipeline endpoint; and a representation of the changes of control parameters which enables collaborative parameter control.

5.3 Pipeline Editor Client

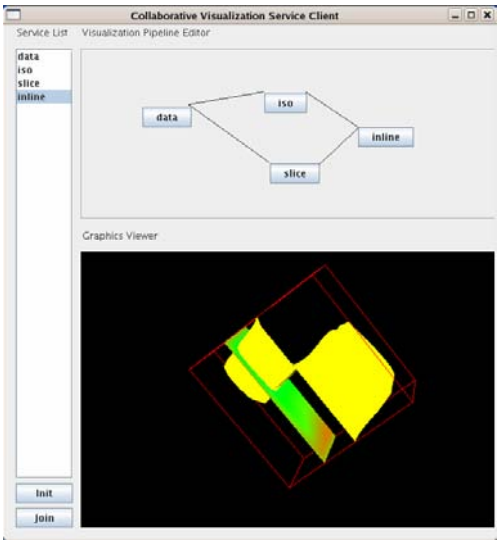


Figure 4 – Pipeline Editor Client

The Pipeline Editor Client (see Figure 4) is initialized with a list of available visualization services shown on the left hand side from an XML format service list file (which makes it possible to retrieve a list of available services from a UDDI server). By clicking on the visualization services, a corresponding instance will be created and displayed as a box in the editor window on the right hand side. The editor window supports drag-and-drop operation. By right clicking on the instance in the editor window, the user can specify the output or input to that instance in order to connect different instances together to create a pipeline. All the participants in the collaboration will see the same pipeline on their editor windows, as they subscribe to the same notification topic – the definition of the current pipeline. At this stage, we simply assume clients know notification

topics published by each visualization service, but in our future work, each visualization service will provide a method which returns a description of notification topics published for the client to subscribe.

5.4 Parameter Control Client

The Parameter Control Client (PCC) provides a user interface for the control parameters of the individual components of the visualization pipeline created by the Pipeline Editor Client. The PCC is implemented using the user interface component from the e-Viz system (Riding et al, 2005), called the e-Viz Client. This component takes an XML description of the visualization pipeline and generates a user interface for each component based on its description. It also provides connectivity from the user interface to the remote visualization component to send and receive parameter values.

The original e-Viz Client used the gViz library as its sole communications mechanism to send/receive parameter changes. This tool has now been extended to allow alternative communications mechanisms to be used in place of the gViz library. This is managed by specifying in the RDF section of the XML description file what mechanism is to be used for each pipeline component. In this case, a Notification Services mechanism has been specified to link the e-Viz Client with the NoCoV system.

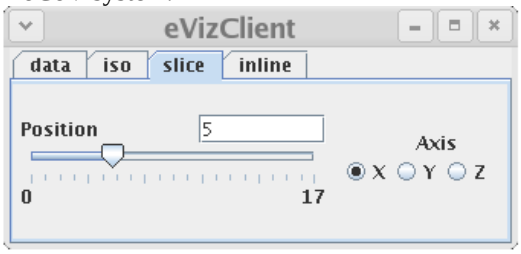


Figure 5: Parameter Control Client for pipeline shown in Figure 4

When a user changes the visualization pipeline using the Pipeline Editor Client, these changes are passed using XML to all of the attached PCCs. The e-Viz Client is designed to respond to snippets of XML that represent alterations to the pipeline and to adjust the user interface accordingly. Thus, when the user adds a module to the pipeline, the e-Viz Client grows a tab to accommodate its control parameter widgets. When a user interacts with a widget on the control panel, these changes are passed to the Pipeline Controller Service which in turn reflects these changes to all PCCs as well as to the intended visualization service. Figure 5 shows the PPC corresponding to the pipeline displayed in Figure 4.

5.5 Simple illustration

We illustrate the use of the NoCoV system with a very simple example. In Figure 6, one user has built a pipeline of two services to visualize a volumetric dataset (*data* – to read in the data; *slice* to display a cross-section).

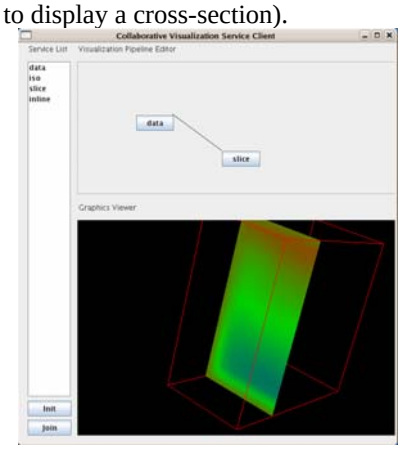


Figure 6 – Slice Visualization

A collaborator then joins the session, and initially sees the same slice, but with ability to view from a different angle. Figure 7 shows the two screens, displayed side-by-side here to show the effect.

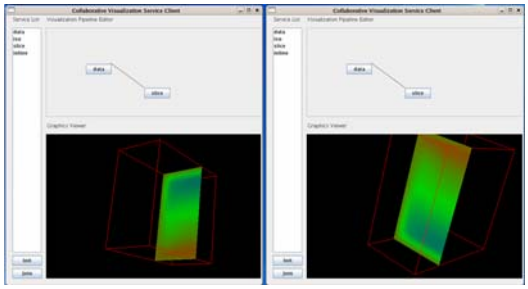


Figure 7 – Collaborative Slice Visualization

The collaborator then suggests that the addition of an isosurface would enhance the understanding of the dataset. They use the pipeline editor to collaboratively alter the pipeline, and the resulting visualizations are shown in Figure 8, again showing the ability of each collaborator to select their own viewing direction.

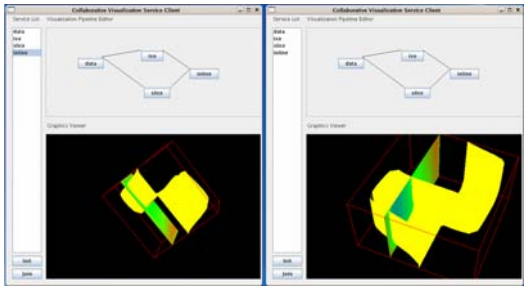


Figure 8 – Adding an Isosurface Service

6. Conclusions and future work

We have presented the vision, design and prototype implementation of a next generation visualization system. It is built using service-oriented concepts, and envisages a worldwide collaboratory in which a research team can come together to visualize data – the collaboration can be at the same time, or at different times. A key feature is exploitation of Grid and Web services technologies, in particular notification services. The publish/ subscribe pattern is used to link the visualization services in a pipeline.

A middle-layer service, the Pipeline Controller Service – acting as a proxy for distributed visualization services - is included to provide collaborative functions for different levels of end-users. Work from the gViz and e-Viz e-science projects is exploited to provide a formal description of the visualization pipeline, and automatically created user interfaces, respectively. A prototype of the proposed system has been implemented as a proof of concept.

The following aspects need to be explored in the next stage to create a comprehensive collaborative visualization system.

Security is an important issue for all collaborative systems. As the prototype is implemented with GT4, the GT4 security mechanisms can be seamlessly applied to NoCoV. The security issues that need to be considered in future work include: setting different roles for users; setting different access permission to each visualization instance in the pipeline; and dynamically changing the valid user list to control the joining and leaving of users in a collaborative session.

The discovery of available visualization services is another important strand. In the prototype, the client gets an available service list from an XML file which contains details of each visualization service – but it is possible to introduce a UDDI server into the system to provide this functionality.

Other issues include the standardization of the data format passed between different types of visualization service, and automatic updating of the Pipeline Controller when new services are brought into the NoCoV repository.

Acknowledgements

Thanks to Stuart Charters (Univ of Durham) for making available an early version of his thesis; and to John Hodrien (Univ of Leeds) for advice on GT4 matters. Jason Wood developed the GUI aspects as part of the EPSRC e-Viz project.

References

Charters, S.M, Holliman, N.S and Munro, M (2004) Visualization on the Grid: a Web Service Approach. Proceedings of the UK e-Science All Hands Meeting, pp202-209.

Charters, S.M (2006) Virtualising Visualisation. PhD thesis, University of Durham.

Duce, D.A, Sagar, M (2005) skML a Markup Language for Distributed Collaborative Visualization. EG UK Theory and Practice of Computer Graphics pp171-178.

GT4 (2006) Globus Toolkit Web site,. http://globus.org/toolkit/

gViz project web site, (2006) http://www.comp.leeds.ac.uk/vis/gviz/

Huang.Y, et al, (2006) WS-Messenger: A Web Services-based Messaging System for Service-Oriented Grid Computing. CCGrid 2006, IEEE Computer Society, pp166-173.

IRIS Explorer web site, (2006) http://www.nag.co.uk/Welcome_IEC.asp

OASIS Web Services Notification TC, (2006). http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn

Publish-Subscribe Notification for Web services, (2006) .http://www.ibm.com/developerworks/library/ws-pubsub/WS-PubSub.pdf

realVNC web site, (2006) http://www.realvnc.com

Riding, M, et al, (2005) e-Viz: Towards an Integrated Framework for High Performance Visualization, Proceedings of the UK e-Science All Hands Meeting, pp1026-1032. See also e-Viz project website, URL: http://www.eviz.org.

Sastry, M and Craig, M (2003) Scalable application visualization services toolkit for problem solving environments. In Proceedings of the UK e-Science All Hands Meeting, pp 520-525.

vtk web site, (2006) http://www.kitware.com.

Walton, J (2004) NAG's IRIS Explorer. In Visualization Handbook, pp 633--654, Academic Press. Available at: http://www.nag.co.uk/IndustryArticles/ch32.pdf

Wood, J, Wright, H and Brodlie, K, (1997) Collaborative Visualization, Proceedings of IEEE Visualization 1997 Conference, edited by R. Yagel and H.Hagen, pp 253-260, ACM Press.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Investigating Visualization Ontologies

Gao Shu[1] **Nick J.Avis[2]** Omer F. Rana[2]

[1]School of Computer Science, Wuhan University of Technology, Hubei 430063, China

[2]School of Computer Science, Cardiff University, U.K

Abstract

The advent of the Grid Computing paradigm and the link to Web Services provides fresh challenges and opportunities for collaborative visualization. Ontologies are a central building block of the Semantic Web. Ontologies define domain concepts and the relationships between them, and thus provide a domain language that is meaningful to both humans and machines. In this paper, the analysis and design of a prototype ontology for visualization are discussed, whose purpose is to provide the semantics for the discovery of visualization services, support collaborative work, curation, and underpinning visualization research and education. Relevant taxonomies are also reviewed and a formal framework is developed.

1.Introduction

Visualization is a powerful tool for analyzing data and presenting results across a wide range of disciplines. Grid computing offers many new opportunity for visualization [19],[20], associated with the marshalling and orchestration of the required physical resources. Perhaps the greatest potential of the Grid with respect to visualization is its capability to support geographically separated teams participating in collaborative visualization efforts [1],[20]. This involves the use of resources needed to produce visualizations, both computing (software system, HPC, networking) and humans (visualization specialization and end user domain experts and users) [2]. The capability to interact with resources that are geographically distributed, as enabled via Grid/eScience infrastructure (such as registry services, remote hosting environments, etc), allows remote services to be accessed and used on demand. Such a computing model assumes that a user is interested in making use of services that are not owned locally and/or cannot be provisioned at their own local site. The viability of such a model has already been demonstrated through the use of numerical services [3] and some emerging Grid-enabled visualization systems [20]. Our present experience of developing Grid enabled visualization services has focused on resource discovery and remote rendering. The Resource Aware Visualization Environment (RAVE) allows adaptation depending on resource availability and load to deliver visualization results to end users in a timely and transparent manner [20]. However, with this experience, we recognize it is imperative to establish common vocabularies and capture and organise visualization domain knowledge to inform and allow the machine-to-machine negotiations necessary for next generation Grid enabled visualization systems. An ontology defines domain concepts and the relationships between them, and thus provides a domain language that is meaningful to both humans and machines. An ontology may also be used to deliver significantly improved (semantic) search and browsing, integration of heterogeneous information sources and improved analytics and knowledge discovery capabilities. In this paper we describe the development of an ontology for visualization. This ontology is designed to provide a common vocabulary for: describing visualization data; processes; products; and support the description and discovery of Web Services;

sharing of process models between visualization developers and users; curation and provenance management of visualization processes and data; and collaboration and interaction between distributed sites offering visualization services [2]. At the same time, Semantic Web Service technologies, such as the Ontology Web Language (OWL), are developing the means by which services can be given richer semantic specifications. Richer semantics can enable more flexible automation of service provision and use, and support the construction of more powerful tools and methodologies. This also makes it possible for users to define their requirements and subsequently connect to services that may be adopted in their work. Ontologies are a central building block of the Semantic Web: they provide formal models of domain knowledge that can be exploited by intelligent agents. A visualization ontology would provide a domain language that is meaningful to humans and machines, describes the configuration of a visualization system and supports the discovery of available Web Services.

We show how a prototype ontology for visualization may be developed using Protégé. It is extendable, and as such, our initial efforts can be viewed as a starting point and catalyst for experts in visualization to create and agree a standard ontology. In Section 2, a summary of related work is provided. Section 3 introduces the Ontology Web Language and Section 4 describes a popular ontology editing tool -- Protégé. The development of ontology for visualization using Protégé will be discussed in the Section 5. Conclusions and future work are presented in Section 6.

2.Related Work

Two workshops have recently been held at the UK's National e-Science Centre (NeSC), one on "Visualization for eScience", held in January 2003 [4] and the other on "Visualization Ontology" held in April 2004, which identified the need to establish an ontology for visualization, and investigated the structure for such an ontology, giving an overall description of what such an ontology should contain [5]. However, there was agreement that these initial efforts were both tentative and incomplete because the creation of an ontology is an iterative activity and the establishment of the ontology for visualization needs consensus within the visualization community itself.

Brodlie at the University of Leeds proposed the notion of a "scientific visualization taxonomy" using the concept of an "E-notation" [6][7], which was first developed at the *AGOCG* visualization meeting in 1992. This is seen as a useful classification and model of the underlying sampled data which may be used for subsequently visualizing this data, however it failed to capture details about how such samples were distributed, or how the visualization was carried out – both fundamental issues in visualization.

Melanie Tory and Torsten Möller in Simon Fraser University have also undertaken significant work on "visualization taxonomies" [8]. Recently, they presented a novel high-level visualization taxonomy. The taxonomy classifies visualization algorithms rather than data. Algorithms are categorized based on the assumptions they make about the data being visualized. As the taxonomy is based on design models, it is more flexible and considers the user's conceptual model, emphasizing the human aspect of visualization [9].

Ed H. Chi at Xerox Parc put forward a new way to taxonomize information visualization techniques by using the Data State Model approach [10]. This research shows that the Data State Model not only helps researchers understand the design space of visualization algorithms, but also helps implementers understand how information visualization techniques can be applied more broadly [11].

Additionally, in an article describing the design space of information visualization techniques, Card and Mackinlay constructed a data-oriented taxonomy [12], which is subsequently expanded in [13]. This taxonomy divides the field of visualization into several subcategories: Scientific Visualization, GIS, Multi-dimensional Plots, Multi-dimensional Tables, Information Landscapes and Spaces, Node and Link, Trees, and Text Transforms. OLIVE is a taxonomy assembled by students in Shneiderman's information visualization class [14], and divides information visualization techniques into eight visual data types: temporal, 1D, 2D, 3D, multi-D, Tree, Network, and Workspace.

3.Ontology Web Language (OWL)

The OWL language is designed for use by applications that need to process the content of information, instead of just presenting information to humans. OWL facilitates greater machine interpretation of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL can be used to explicitly represent the meaning of terms in vocabularies and define the relationships between those terms. OWL is part of the growing stack of W3C recommendations related to the Semantic Web. Compared with XML, XML-Schema, RDF, and RDF-Schema, OWL adds additional vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointedness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. OWL has three increasingly-expressive sub-languages: OWL Lite, OWL DL, and OWL Full [15]. Our ontology for visualization is represented in OWL DL-- because OWL DL is much more expressive than OWL-Lite and is based on Description Logics, it is therefore possible to automatically compute a classification hierarchy and check for inconsistencies in an ontology that conforms to OWL-DL.

4.Building OWL Ontology with Protégé

An OWL ontology can be regarded as a network of classes, properties, and individuals. Classes define names of the relevant domain concepts and their logical characteristics. Properties (sometimes also called slots, attributes or roles) define the relationships between classes, and allow the assignment of primitive values to instances. Individuals are instances of the classes with specific values for the properties. Our visualization ontology (called here "VO" for short) is developed using Protégé_3.1_beta with OWL-plugin. Protégé is an open platform for ontology modeling and knowledge acquisition. The OWL Plugin [18] can be used with Protégé, and enables a user to load and save OWL files in various formats, to edit OWL ontologies with custom-tailored graphical widgets, and to provide access to reasoning based on description logic. The OWL Plugin user interface provides various default tabs for editing OWL classes, properties, forms, individuals, and ontology metadata.

As an extension of Protégé, the OWL Plugin has a large and active user community, a library of reusable components, and a flexible architecture. The OWL Plugin therefore has the potential to become a standard infrastructure for building ontology-based Semantic Web applications [16].

5.Ontology for visualization

5.1 The overview of VO

A taxonomy is a good mental starting point for building an ontology. Unfortunately, there has not been a universally, well-accepted taxonomy for visualization developed as yet. In our opinion, the ontology for visualization must be a compromise between function, understandability and uniformity. As a starting point therefore, we

synthesize some existing taxonomies, mainly based on Ken Brodlie [6] and Melanie Tory, Torsten Möller's [8] taxonomies, and present these in an organized structure which highlights the connection between data models, visualization techniques and the data representation. Taking the data model for example, firstly, it is categorized into two types: discrete model and continuous model, and then the continuous model is classified according to the type of each variable: scalar, vector, tensor, point or multivariate, and each of which is broken down further according to the dimensionality and number of variables, and is denoted by the E-notation [6]. The factor of time is neglected here, and the relationship between the data model and the corresponding visualization techniques should be considered mainly in the process of category. The discrete model is classified into either a connected or unconnected model, and the unconnected model is broken down further according to dimensionality of the data involved, each of which is also denoted by the E-notation. A more detail description is presented in Section 5.3.

Our VO is a web-accessible and was envisioned to be a prototype ontology used to enable automatic composition of Web-based visualization services. So we do not attempt to capture all knowledge about the visualization domain, rather our main goal is to cover as many concepts and their relations as possible to provide machine-readable formal specifications for the discovery of visualization services. Moreover, some of decisions made in the process of building VO were not rigorously justified but rather based on the authors' intuition and sometimes directed by the limitations of the tools currently available.

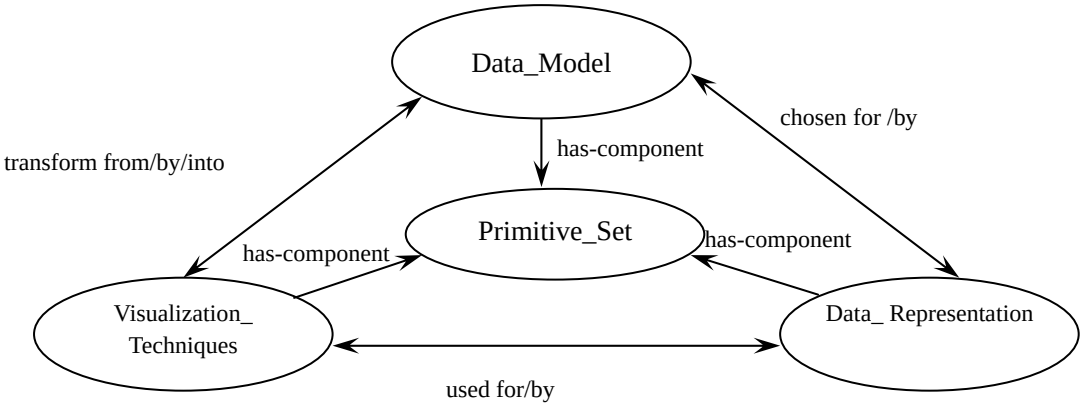


Figure 1 Relationship between the four main classes in the VO

The VO consists of classes, properties and individuals. Classes are interpreted as sets that contain individuals. They are described using formal descriptions that state precisely the requirements for membership of the class. Properties illustrate relationships between two individuals. And individuals represent objects in the domain that we are interested in.

The VO has four abstract classes representing the main concepts in the visualization domain: Data_Model, Visualization_Techniques, Data_Representation, and Primitive_Set. Their relationship is shown in figure 1.

Data_Model describes the user's data model, Visualization_Techniques defines a variety of techniques and algorithms used to transform or visualize the data model, Data_Representation contains multimodal attributes, which enables a user to chose an appropriate representation, whereas Primitive_Set includes the elements used as the building block of above classes.

5.2 Primitive Set class

The subclasses of Primitive_Set class are the basic concepts which might be linked through *has-* property with corresponding classes. In a sense they serve as a set of basic concepts upon which the ontology is built. Figure 2 shows the portion of Primitive_Set as corresponding to blocks for the Data_Model.

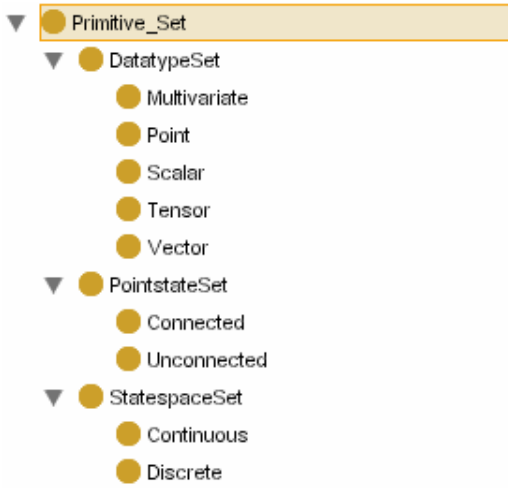


Figure 2. The contents of the class Primitive_Set

To create any ontology, it is necessary to provide concise definitions of the base concepts involved. It is often the case, however, that some of these concepts come from different fields and it is better to have them formalized by experts in these fields. VO may need some fundamental mathematical concepts to be included. One way to do this is to link the ontology to another already developed ontology. For example, the Mathematics on the Net [MONET 2004][17] project is developing OWL-based mathematical ontologies using the OpenMath and MathML initiatives. However, at the moment, since they are still being refined, we hope to investigate the option of using them at a later time, so VO defines some simple mathematical concepts on its own. But the approach seems to be most promising and is likely to be used in the future when ontologies from base knowledge domains are finalized.

5.3 Data_Model class

The hierarchies within the Data_Model class, which are shown in figure 3, are basically the same as our taxonomy for visualization. However, we can go further with an ontology.

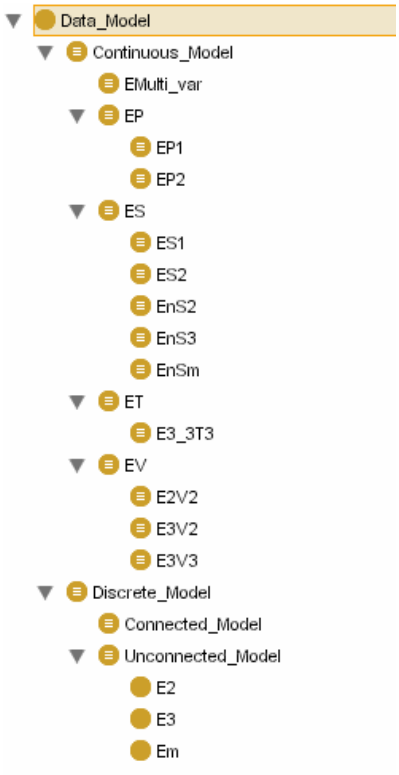


Figure 3. The content of Data_Model

Where: EnSm denotes a (n) scalar entity on a m-dimensional domain.

EnVm denotes an entity has (n-)vector data on a m-dimensional domain.

En_nTm denotes an entity has a (n;n) tensor data on a m-dimensional domain(where n=3).

EPm denotes a m-dimensional domain of point-data (where n=1,2).

EMulti_var denotes an entity has multi-dependent variables.

Em denotes an entity has discrete and unconnected data on m-dimensional domain.

We can put more complex and meaningful machine understandable knowledge into the VO by using stronger semantic connections in the OWL-based ontology: OWL is used to relate classes through simple superclass/subclass relationships and properties, and also

distinguishes between necessary (inherited) conditions (superclasses) and necessary and sufficient conditions (equivalent classes). Furthermore, an OWL-based description allows users to restrict classes or related classes with description logic symbols, such as $\forall, \exists, \neg, \cap, \cup$, where:

$\forall$ means allValuesFrom,

$\exists$ someValuesFrom,

$\neg$ complementOf,

$\cap$ intersectionOf,

$\cup$ unionOf.

At the same time, one of the key features of an ontology that are described using OWL-DL is that they can be processed by a reasoner, such as RACER in Protégé. This allows checks for inconsistencies, hidden dependencies, redundancies, and misclassification. Taking E3V3 as an example, we define its necessary & sufficient condition as *EV*, *has-Dimension = 3*, *has-Vector = 3*, which mean that E3V3 is the subclass of EV, the number of its vector argument equals 3 and dimension is 3. We can also define that its necessary condition is $\forall$ *is-transformed-by* (AnVm $\cup$ A3V3), where n , m=2,3, which means it can be transformed or visualized by algorithms included in classes AnVm and A3V3, which relates this data model to the corresponding algorithm. Meanwhile, it inherits restriction condition $\forall$ *has-DatatypeSet Vector* and $\forall$ *has-StatespaceSet Continuous* from its superclass EV, which is shown in figure 4. Through these restrictions on properties and logic statements, our VO can provide Semantic Web agents with background knowledge about domain concepts and their relationships. This knowledge can be exploited in various ways, for example to drive context-sensitive search functions, i.e. agents can use their ontological knowledge to match a user data model with the available visualization services offering the corresponding algorithms [21]. Taking EnS3 as an example, if the user's data model is EnS3, the matchmaking agent can search the services which offer the algorithms, included in the class AnS3 shown in figure 5, such as Marching_Cube or Contour_Connecting and so on, to visualize the data model. All of these can be done by means of EnS3's necessary condition $\forall$ *is-transformed-by AnS3*.

5.4 Visualization_Techniques class

The subclasses of Visualization_Techniques

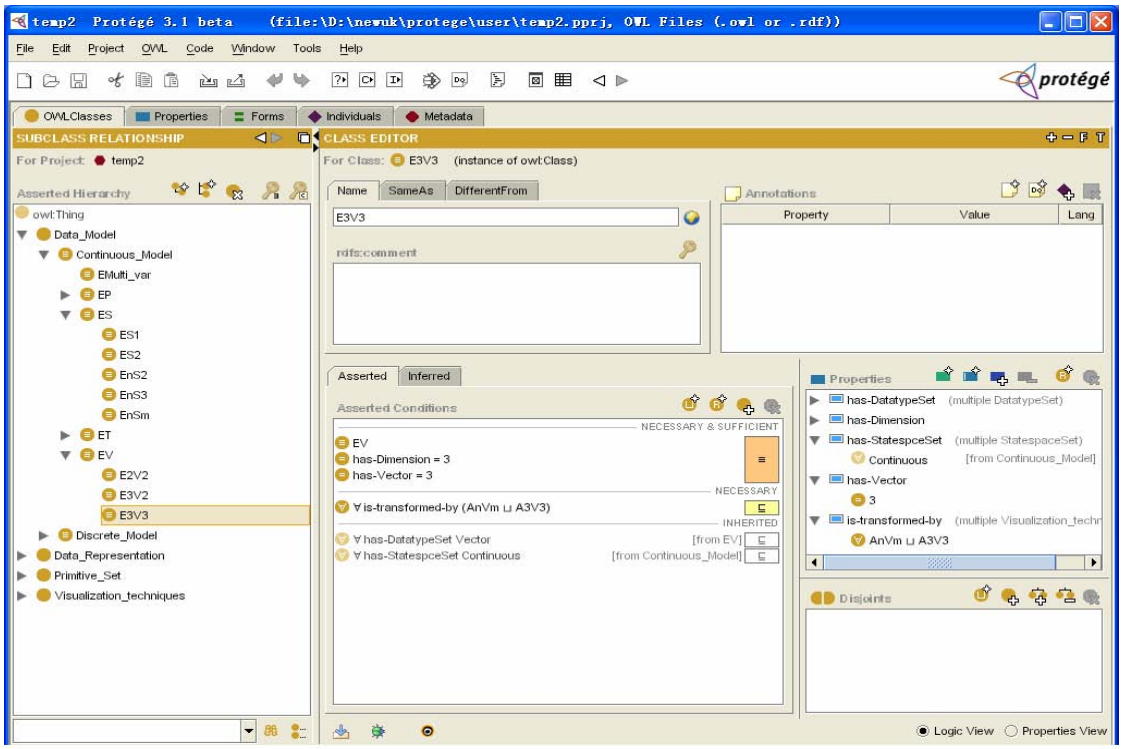


Figure 4 The definition of the class E3V3 in Protégé OWL plugin

define a set of techniques and algorithms which can be used to transform or create visual representations of data using a data model. So techniques/algorithms are categorized based on the data model rather than on the data itself, which facilitates the transformation or visualization of the data model. The relationship between techniques/algorithms and data model is achieved by three properties: *is-transformed-by*, *transform-from and transform-into-model*. The property *transform-from* is an inverse property of *is-transformed-by*. The former means an algorithm can be used to visualize the corresponding data model, the latter the inverse. The range of the property *transform-into-model* is Visualization_Techniques, which means we can use an algorithm to transform a data model into a simpler form. It can therefore be seen that it is possible to match data for specific applications to the most appropriate visualization techniques/algorithms by means of the above three properties. A fragment of subclass of Visualization_Techniques is shown in figure 5 (for example, AS2 denotes that its subclasses can be used to transform or visualize the data model ES2).

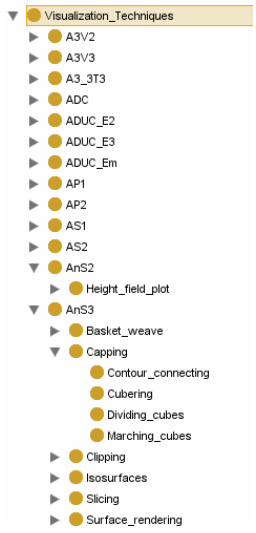


Figure 5. The content of Visualization_Techniques

5.5 Data_ Representation class

The reason that Data_Representation class should be included is that some models of representation are specific to classes of data, for example work in flow visualization and graph visualization have distinct categories of representations. So users should be allowed to make the appropriate representation choice based on their information need. Its subclasses contain multimodal attributes, i.e. visual, haptic, sonic, olfactory, physical attribute etc. The visual attribute is further split into four subclasses spatialization, timing, color and transparency [7].

6.Conclusion and future work

In this paper, we have described our experience of building an ontology for visualization, although it is accepted that this is both incomplete and tentative. Our VO is designed as an OWL-based prototype ontology whose purpose is to provide a vocabulary by which users and visualization systems can communicate, and it is being used in matchmaking portal for discovery of visualization services [21]. Several specific directions for future work include:

- To populate the VO in terms of Operators, Techniques, Algorithms and (low-level) Transformation with respect to the Data Model.
- To determine the limits of the expressiveness of the VO.
- To extend the above VO to include a user model and actions.
- To extend the above VO to include temporal aspects/conditions.
- To determine if the VO can be used to characterize and describe lower levels of details such as NAG-EXPLORER meshes using the same constructs.
- To create a system which uses the VO to automatically build ALL permissible

visualizations given knowledge of the data.

References

1. Brodlie, K. W., Duce, D. A., Gallop, J. R., Walton, J. P. R. B. & Wood, J. D. Distributed and Collaborative Visualization.Computer Graphics Forum 23 (2), 223-251,2004
2. D. J. Duke, K.W. Brodlie and D. A. Duce, Building an Ontology of Visualization
3. Simone Ludwig, O. F. Rana, J. A. Padget and W. Naylor, Matchmaking framework for Mathematical Web Services, Journal of Grid Computing, 2006.
4. National e-Science Centre. Visualization ontologies. http://www.nesc.ac.uk/talks/393/vis ontology report.pdf.
5. National e-Science Centre. Visualization for e-science. http://www.nesc.ac.uk/esi/events/130/workshop report.pdf.
6. K.W. Brodlie, 1992.Visualization Techniques, in Scientific Visualization - Techniques and Applications, edited by K.W. Brodlie, L.A. Carpenter, R.A. Earnshaw, J.R. Gallop, R.J.Hubbold, A.M. Mumford, C.D. Osland and P. Quarendon, Chapter 3, pp 37-86, Springer-Verlag.

7.K.W. Brodlie, 1993. A Classification Scheme for Scientific Visualization, in Animation and Scientific Visualization, edited by R. A. Earnshaw and D. Watson, pp 125-140, Academic Press.

8. Melanie Tory and Torsten Moller. A model-based visualization taxonomy. Technical Report SFU-CMPT-TR2002-06, Computing Science Dept., Simon Fraser University, 2002.
9. Melanie Tory and Torsten Möller. Rethinking Visualization: A High-Level Taxonomy, IEEE Symposium on Information Visualization, pp. 151-158, Oct. 2004.
10. Ed H. Chi and J. T. Riedl. An Operator Interaction Framework for Visualization Systems. Symposium on Information Visualization (InfoVis '98), Research Triangle Park, North Carolina: 63-70, 1998.
11. Ed H. Chi. A Taxonomy of Visualization Techniques using the Data State Reference Model. In Proceedings of the Symposium on Information Visualization (InfoVis '00), pp. 69--75. IEEE Press, 2000. Salt Lake City, Utah.
12. S. K. Card, J. D. Mackinlay. The Structure of the Information Visualization Design Space. Proceedings of IEEE Symposium on Information Visualization (InfoVis ’97), Phoenix, Arizona, 92-99 Color Plate 125, 1997.
13. S. K. Card, J. D. Mackinlay and B. Shneiderman. Information Visualization: Using Vision to Think. Morgan-Kaufmann, San Francisco, California, 1999.
14. OLIVE: On-line Library of Information Visualization Environments. http://otal.umd.edu/Olive/, 1999.
15. OWL Web Ontology Language Overview : http://www.w3.org/TR/2004/REC-owl-features-20040210/
16. Holger Knublauch, Ray W. Fergerson, Natalya F. Noy, Mark A. Musen. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. Third International Semantic Web Conference - ISWC 2004
17. MONET Consortium. MONET Home Page, www. Available from http://monet.nag.co.uk.

18 Holger Knublauch, Mark A. Musen, and Alan L. Rector.Editing description logics ontologies with the Prot´eg´e OWL plugin. In International Workshop on Description Logics, Whistler, BC, Canada, 2004.

19 I J Grimstead, D W Walker and N J Avis. Resource Aware Visualization Environment – RAVE, accepted for publication in Concurrency and Computation: Practice and Experience.

20 Ian J. Grimstead, David W. Walker, Nick J. Avis. Collaborative Visualization: A Review and Taxonomy. In Ninth IEEE International Symposium on Distributed Simulation and Real-Time Applications, 2005, pp. 61-69.

21 Gao Shu Omer F. Rana, Nick J Avis and Chen Dingfang. Ontology_based Semantic Matchmaking Approach. Accepted of publication.

Proceedings of the UK e-Science All Hands Meeting 2006, EPSRC (2006), ISBN 0-9553988-0-9

Meshing with Grids: Toward functional abstractions for grid-based visualization

Rita Borgo* David Duke† Malcolm Wallace‡ Colin Runciman§
University of Leeds University of Leeds University of York University of York

Abstract

A challenge for grid computing is finding abstractions that separate concerns about what a grid application must achieve from the specific resources on which it might be deployed. One approach, taken by a range of toolkits and APIs, is to add further layers of abstraction above the raw machine and network, allowing explicit interaction with grid services. The developers' view, however, remains that of a stateful von Neumann machine, and the interleaving of grid services with domain computations adds complexity to systems.

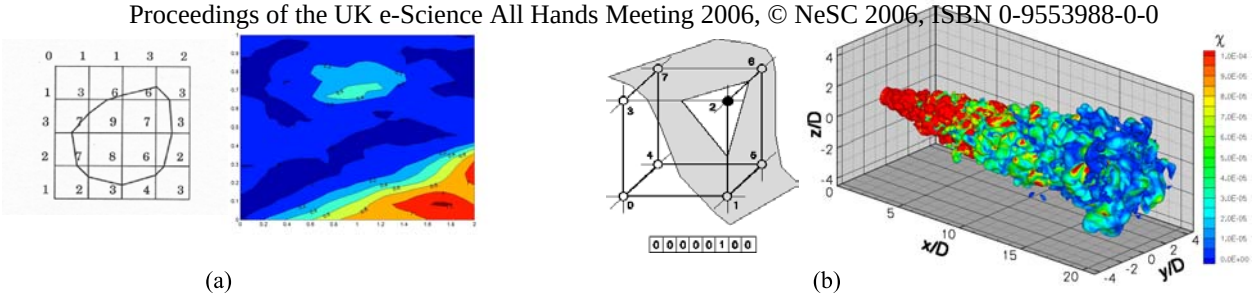
An alternative is to replace this tower of abstraction with a new kind of programming structure, one that abstracts away from complexities of control and location while at the same time adapting to local resources. This paper reports early results from a project exploring the use of functional language technologies for computationally demanding problems in data visualization over the grid. Advances in functional language technology mean that pure languages such as Haskell are now feasible for scientific applications, bringing opportunities and challenges. The opportunities are the powerful types of 'glue' that higher order programming and lazy evaluation make possible. The challenge is that structures that work well within stateful computation are not necessarily appropriate in the functional world. This paper reports initial results on developing functional approaches to an archetypal visualization task, isosurfacing, that are a preliminary step towards implementing such algorithms on a polytypic grid.

1 Introduction

Distribution and parallelism are inherent properties of grid computing environment, and grid programming requires attitude and skills that go beyond that of traditional sequential or even parallel and distributed programming. Programmers need to cope not only with sharing of resources but to handle computation in an environment characterised by heterogeneous and dynamic resources. Resources allocated to a program may vary between executions, and in some cases may change during execution. Managing this complexity ideally requires a combination of run-time support systems and high level abstraction that allow the programmer to separate cleanly applications concerns and grid interfaces. Building such layers of service abstraction is an approach that has served computing well in the past, giving developers reusable domain-independent blocks for building an application. For example middleware and libraries such as Globus, OpenGL, and VTK are likewise abstractions that provide high-level access to lower-level services (HPC tasks, graphical rendering, and visualization, respectively).

Data visualization is an application domain that has a close affinity with grid computing for two reasons: large scale datasets are challenging computational problem, and visualization is frequently an important component of grid computing applications. A number of architectures have been proposed and developed for data visualization, including spreadsheets, relational databases, spray-rendering, scene graphs, and pipelines. They provide a layer of application-oriented services on which problem-specific visualization tools can be constructed. Although some of these architectures can be used in a grid environment (e.g. Cactus [1]) this is by explicit use of low-level services. Of the approaches explored to date, the pipeline model has found the most widespread use. It underlies the implementation of well-known systems such as AVS, SCIRun, and also serves as a conceptual model for visualization workflow as VTK [16],. For the pipeline model, services provide the capability to organize visualization operations within a dataflow-like network. Some pipelined systems extend the basic model with demand-driven evaluation and streaming of dataset chunks, again frozen into the service layer. *Streaming* [12] is an enrichment to the basic model that allows a pipeline to pass datasets in chunks. For scientific data, such chunks are usually spatially contiguous subsets of the full extent. Some algorithms, for example Marching Cubes [14], can operate on individual chunks in isolation. Others require access to the full dataset, for example surface reconstruction: the dataset may be passed as a sequence of chunks, with algorithms working downstream and upstream on different sequencesof the pipeline. However, this layered approach fixes design decisions associated with the services, without regard for the operations that are implemented in terms of those services. Pipeline services provide a lazy, dataflow-like model, but client operations are defined as a separate layer of stateful computation.

While pipeline capabilities have advanced, both the services and the algorithms that use those services continue to be implemented using imperative languages, usually C or C++. The underlying computational model



(a) (b)

Figure 1: (a) Marching cubes applied to a 2D dataset (e.g. marching squares) and extracted isocontour; (b) Marching cubes applied to a 3D dataset and extracted isosurface.)

is *call-by-value* parameter-passing, yet the way to access services from an application is conceptually call-by-need. In contrast, non-strict functional languages such as Haskell [7] use a call-by-need evaluation strategy in which function arguments are only evaluated to the extent they are demanded (if at all). Apart from closely matching the pipeline model, this strategy also provides a 'new kind of glue' [8] for assembling programs from components. Recent work on functional languages has produced new forms of generic programming including polytypic functions (GH [3]) and type-generic frameworks (SYB [10]).

In Generic Haskell (GH), a polytypic function captures a family of polymorphic functions in a single, inductive, and typed definition. Instances of the family for specific types can be generated automatically by the GH compiler. In a grid context, polytypic definitions may support single functions that capture patterns of computation over a family of data representations.

This paper sets out initial findings on work aimed at using functional technologies for data visualization over the grid. Implementations of 'lazy' functional languages have advanced significantly over the last decade. They now have well-developed interfaces to low-level services such as graphics and I/O. In this paper we take surfacing as an archetypal visualization task. We illustrate how pipelining and demand-driven evaluation become naturally integrated within the expression of an algorithm. The resulting implementations have a pattern of space utilization quite different to their imperative counterparts, occupying an intermediate point between purely in-core and out-of-core approaches.

Section 2 revisits the basic marching cubes algorithm for surface extraction, using the lazy functional language Haskell [7]. Through a series of refinements, we show how pipelining and demand-driven evaluation allow the use of *memoization* to improve performance. Section 2.5, on evaluation, gives particular attention to the space performance of our implementation. The lazy streaming approach set out here features low memory residency, even with larger datasets. Section 3 discusses related research. The work reported here is a first step in a much larger programme of work, and in Section 4 we summarize the achievement to date.

2 Marching Cubes, Functionally

Without giving a full tutorial on Haskell, we need to introduce some key aspects of functional languages, for which we use the classic Marching Cubes algorithm as an exemplar. Marching Cubes is a technique widely used in the visualization field to represent 3D scalar fields in terms of approximating polygonal surfaces. Given a scalar field $\mathscr{F}$ and a constant value c then the locus of points x, with $x \in \mathscr{R}^3$, that satisfies the equation $\mathscr{F}(x) = c$ represents an isosurface of value c. Given a threshold value c the marching cube algorithm proceeds through the scalar field, testing the corner of each voxel (or cube) in the scalar field as being either above or below the threshold. If one or more corners of the cube have values less than the threshold, and one or more have values greater than this value, the voxel must contribute some component of the isosurface. Locating which edges of the cube are intersected by the isosurface, it is possible to determine the polygons that must be created to represent the part of isosurface that is intersected by the voxel. The result of the marching cubes algorithm is a smooth surface that approximates the isosurface that is constant along a given threshold. Figure 1 represents the algorithm for the 2D and 3D case respectively. Examples of 2D scalar fields are temperature, pressure and humidity in a meteorologichal context, examples of 3D scalar fields can be taken from the medical field lik eCT scans of human skull or body parts.

We first implement it in the standard fashion, iterating through an array of sample values, then refine the implementation into suite lazily streaming variations. These illustrate two of the main benefits of laziness – on-demand processing (permitting fine-grained pipelining of input and output data), and automatic sharing of already-computed results. In the following section we introduce some of the early results presented in more detail in [5] we then exploit how our approach can interestingly suit within the grid context.

2.1 Ordinary, array-based algorithm.

First, we explore a straightforward representation of the dataset as a three-dimensional array of sample values.

```
type XYZ = (Int,Int,Int)
```

```
type Num a => Dataset a = Array XYZ a
```

These `type` definitions declare synonyms for the actual array representation. Concrete type names are capitalised, for instance the `Array` index domain type is `XYZ`. The type variable (lower-case `a`) in the range of the array indicates that the type of the samples themselves is generic (polymorphic). The predicate `Num a` constrains the polymorphism: samples must have arithmetic operations defined over them. Thus, we can reuse the algorithm with bytes, signed words, floats, complex numbers, and so on, without change.

```
isosurface :: Num a => a -> Dataset a
                        -> [Triangle b]
```

This type declaration (signature) of the Marching Cubes `isosurface` function shows that it takes two arguments, a threshold value and the dataset, and computes from them a sequence of triangles approximating the surface. The triangles can be fed directly into e.g. OpenGL for rendering. The full visualization pipeline can be written:[1]

```
pipeline t =
    mapper view . normalize . isosurface t
                . reader
```

Here the dot `.` operator means pipelined composition of functions. The last function in the chain is applied to some input (a filename), and its results are fed back to the previous function, whose results are fed back, and so on. Since this operator is the essence of the pipeline model, let's look briefly at its definition:

```
(.) :: (b->c) -> (a->b) -> a -> c
(f . g) x  = f (g x)
```

Dot takes two functions as arguments, with a third argument being the initial data. The result of applying the second function to the data is used as the argument to the first function. The type signature should help to make this clear - each type variable, `a`, `b`, and `c`, stands for any arbitrary (polymorphic) type, where for instance each occurrence of `a` must be the same, but `a` and `b` may be different. Longer chains of these compositions can be built up, as we have already seen in the earlier definition of `pipeline`. Dot is our first example of *higher-order* function. From the very name "functional language" one can surely guess that functions are important. Indeed, passing functions as arguments, and receiving functions as results, comes entirely naturally. A function that receives or returns a function is called higher-order.

Shortly, we will need another common higher-order function, `map`, which takes a function `f` and applies it to every element of a sequence:

```
map :: (a->b) -> [a] -> [b]
map f []     = []
map f (x:xs) = f x : map f xs
```

This definition uses pattern-matching to distinguish the empty sequence `[]`, from a non-empty sequence whose initial element is `x`, with the remainder of the sequence denoted by `xs`. Colon `:` is used both in pattern-matching, and to construct a new list.

Now to the algorithm itself. We assume the classic table, either hard-coded or generated by the Haskell compiler from some specification. Full details of these tables are not vital to the presentation and are omitted; see [14] for example.

Marching Cubes iterates through the dataset from the origin. At every cell it considers whether each of the eight vertices is below or above the threshold, treating this 8-tuple of Booleans as a byte-index into the case table. Having selected from the table which edges have the surface passing through them, we then interpolate the position of the cut point on each edge, and group these points into threes as triangles, adding in the absolute position of the cell on the underlying grid.

```
isosurface threshold sampleArray =
    concat [ mcube threshold lookup (i,j,k)
           | k <- [1 .. ksz-1]
           , j <- [1 .. jsz-1]
           , i <- [1 .. isz-1] ]
  where
    (isz,jsz,ksz)  = rangeSize sampleArray
    lookup (x,y,z)
           = eightFrom sampleArray (x,y,z)
```

In Haskell, application of a function to arguments is by juxtaposition so in the definition of `isosurface`, the arguments are `threshold` and `sampleArray`. The standard array function `rangeSize` extracts the maximum co-ordinates of the grid.

The larger expression in square brackets is a list *comprehension*[2], and denotes the sequence of all applications of the function `mcube` to some arguments, where the variables `(i,j,k)` range over (or are *drawn from*) the given enumerations. The enumerators are separated from the main expression by a vertical bar, and the evaluation order causes the final variable `i` to vary most rapidly. This detail is of interest mainly to ensure good cache behaviour, if the array is stored with x-dimension first. The comprehension can be viewed as equivalent to nested loops in imperative languages.

The result of computing `mcube` over any single cell is a sequence of triangles. These per-cube sequences are concatenated into a single global sequence, by the standard function `concat`.

Now we look more closely at the data structure representing an individual cell. For a regular cubic grid, this is just an 8-tuple of values from the full array.

```
type Cell a = (a,a,a,a,a,a,a,a)

eightFrom :: Array XYZ a -> XYZ -> Cell a
eightFrom arr (x,y,z) =
    ( arr!(x,y,z),       arr!(x+1,y,z)
```

[1] Our Haskell implementation is actually built directly on the HOpenGL binding, so the mapping phase is implemented slightly differently, via a function that is invoked as the GL display callback.

[2] It bears similarities to Zermelo-Frankel (ZF) set comprehensions in mathematics.

```
    , arr!(x+1,y+1,z),   arr!(x,y+1,z)
    , arr!(x,y,z+1),     arr!(x+1,y,z+1)
    , arr!(x+1,y+1,z+1), arr!(x,y+1,z+1)
    )
```

Finally, to the definition of `mcube`:

```
mcube :: a -> (XYZ->Cell a) -> XYZ
          -> [Triangle b]
mcube thresh lookup (x,y,z) =
    group3 (map (interpolate
                   thresh cell (x,y,z))
                (mcCaseTable ! bools))
  where
    cell  = lookup (x,y,z)
    bools = toByte (map8 (>thresh) cell)
```

The `cell` of vertex sample values is found using the `lookup` function that has been passed in. We derive an 8-tuple of booleans by comparing each sample with the threshold (`map8` is a higher-order function like `map`, only over a fixed-size tuple rather than an arbitrary sequence), then convert the 8 booleans to a byte (`bools`) to index into the classic case table (`mcCaseTable`).

The result of indexing the table is the sequence of edges cut by the surface. Using `map`, we perform the interpolation calculation for every one of those edges, and finally group those interpolated points into triples as the vertices of triangles to be rendered. The linear interpolation is standard:

```
interpolate :: Num a => a -> Cell a -> XYZ
                -> Edge -> TriangleVertex
interpolate thresh cell (x,y,z) edge =
    case edge of
      0 ->  (x+interp,   y,  z)
      1 ->  (x+1, y+interp,  z)
      ...
      11 -> (x,   y+1,  z+interp)
  where
    interp = (thresh - a) / (b - a)
    (a,b)  = selectEdgeVertices edge cell
```

Although `interpolate` takes four arguments, it was initially applied to only three in `mcube`. This illustrates another important higher-order technique: a function of n arguments can be *partially applied* to its first k arguments; the result is a specialised function of $n-k$ arguments, with the already-supplied values 'frozen in'.

2.2 Factoring Common Design Patterns

The implementation outlined so far is a naive and quite straightforward "functional translation" of the traditional marching cubes algorithm. As for the original C implementation sections where improvements can be made are easy to spot: (1) the *entire dataset* is assumed to be loaded in memory; (2) common behaviours like threshold comparison and interpolant computation are *not factored* out and *shared* between adjoining cells; (3) and ambiguous cell configurations are *not considered*.

The ideal solution to the listed issues would intuitively be an implementation of the algorithm that would take care of all the three kind of problems simultaneously. However often ideal solutions are taylored to specific resources of computation, policy not affordable in a multi-facet environment like the grid. Experience shows how to one holistic solution it is often preferable an amenable suite of solutions capable to efficiently provide optimal trade-offs between results and available computational power: different implementations for different environments. However the ability to swap between different versions of the same algorithms involves evincing crucial patterns of behaviour within the algorithm itself. Exploiting the abstraction and expression power of Haskell in the context of the marching cubes algorithm, we have outlined several different implementations that in turn take care of memory issues, when the entire dataset cannot be loaded in memory, and sharing of computation, when the same computation is carried on the same data more than once.

2.2.1 Stream based algorithm

When dealing with large dataset the monolithic array data structure presented so far is not feasible; it simply may not fit in core memory. A solution is to separate traversal and processing of data. For its inner structure the marching cube algorithm only ever needs at any one moment, a small partition of the whole dataset: a single point and 7 of its neighbours suffices, making up a unit cube of sample values. If we compare this with a typical array or file storage format for regular grids (essentially a linear sequence of samples), then the unit cube is entirely contained within a "window" of the file, corresponding to exactly one plane + line + sample of the volume. The ideal solution is to slide this window over the file, constructing one unit cube on each iteration, and dropping the previous unit cube.

Figure 2 illustrates the idea.

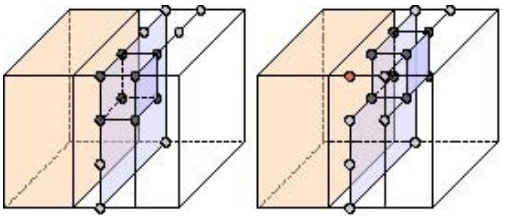


Figure 2: Sliding a window over a grid

Following this idea our first implementation provides a streamed version of the algorithm as follow:

```
isosurfaceS :: (Ord a, Int a, Fract b) =>
            a -> Dataset a -> [Triangle b]
isosurfaceS thresh (D size samples) =
    concat (zipWith2 (mcubeS thresh)
              (cellStream samples) allXYZ )
 where
  cellStream = disContinuities size .
                                mkStream size
  allXYZ  = [XYZ i j k | k <- [0 .. ksz-2]
```

```
                        , j <- [0 .. jsz-2]
                        , i <- [0 .. isz-2]]
  (XYZ isz jsz ksz) = size

mcubeS :: (Ord a, Int a, Fract b) => a ->
             Cell a -> XYZ -> [Triangle b]
mcubeS thresh cell xyz =
 group3 (map (interpolate thresh cell xyz)
               (mcCaseTable ! byte))
 where
  byte = toByte (mapCell (>thresh) cell)
```

Haskell allows us to read data out of a file in this streamed fashion using lazy file I/O. The content of the file appears to the program as a sequence of bytes, extended on-demand one byte at a time.[3] As for dropping data after it has been consumed, Haskell is a garbage-collected language, so when a datum is no longer referenced by the computation, the memory storing it is recycled automatically.

The datatype representing the dataset is constructed from a lazy sequence of samples, stored along with the bounds of the grid:

```
data Num a => Dataset a = D XYZ [a]
```

The sliding window of eight point values (cell) is extracted from the lazy stream of samples as follows: 8 copies of the datastream are laid side-by-side, one value from each of the 8 is then repeatedly sliced off and glued together into a cell. We refer to [5] for a more detailed description of the code. Table 2 shows the memory performances of our streamed implementation.

The advantage of call-by-need over call-by-name is that although the evaluation of an item might be delayed until it is needed, it is never repeated, no matter how often the value is used. If we want to share a computation between different parts of the program, we just arrange for the shared value to be constructed in one place, by one expression, rather than constructing it multiple times which leads to multiple evaluations.

In the streaming version of marching cubes presented so far, we can see that the reading of sample values from file is shared and performed only once. However, by construction, comparison against the threshold value (in `mcubeS`) is performed eight times for every sample, because on each occasion, the sample is at a different vertex position in the cell. Depending on the computational power available it is sometime worth to allow for redundant computations rather than to increase the complexity of the algorithm or adding extra structure at memory expenses. However such a conclusion can be often achieved only after a performance comparison of both solutions. For what concerns algorithm complexity in our implementation it does not represents an issue, to compute the comparison only once per sample, we just need to do the thresholding against the original byte-stream *before* it is tupled up into cells, rather than after.

```
isosurfaceT :: (Ord a, Int a, Fract b) =>
           a -> Dataset a -> [Triangle b]
isosurfaceT thresh (D size samples) =
    concat (zipWith3 (mcubeT thresh)
             (cellStream samples)
             (idxStream samples) allXYZ )
 where
  cellStream = disContinuities size .
                                mkStream size
  idxStream  = map toByte . cellStream .
                                map (>thresh)
  allXYZ  = [XYZ i j k | k <- [0 .. ksz-2]
                       , j <- [0 .. jsz-2]
                       , i <- [0 .. isz-2]]
  (XYZ isz jsz ksz) = size

mcubeT :: (Int a, Fract b) => a -> Cell a
           -> Byte -> XYZ -> [Triangle b]
mcubeT thresh cell index xyz =
 group3 (map (interpolate thresh cell xyz)
             (mcCaseTable ! index))
```

Taking the notion of sharing-by-construction one step further, we now memoize the interpolation of edges. Recall that, in the result of the `mcCaseTable`, the sequence of edges through which the isosurface passes may have repeats, because the same edge belongs to more than one triangle of the approximated surface. But in general, an edge that is incident on the isosurface is also common to four separate cells, and we would like to share the interpolation calculation with those cells too. So, just as the threshold calculation was performed at an outer level, on the original datastream, something similar can be done here building 12-tuple of possible edges, one entry for each cube edge, adding a per-edge description of how to compute the interpolation

```
type CellEdge a = (a,a,a,a,a,a,a,a,a,a,a,a)

isosurfaceI :: (Ord a, Int a, Fract b) =>
           a -> Dataset a -> [Triangle b]
isosurfaceI thresh (D size samples) =
    concat (zipWith3 mcubeI
             (edgeStream samples)
             (idxStream samples) allXYZ )
 where
  edgeStream = disContinuities size .
                    mkCellEdges thresh size
  cellStream = disContinuities size .
                                mkStream size
  idxStream  = map toByte . cellStream .
                                map (>thresh)
  allXYZ  = [XYZ i j k | k <- [0 .. ksz-2]
                       , j <- [0 .. jsz-2]
                       , i <- [0 .. isz-2]]
  (XYZ isz jsz ksz) = size
```

[3] For efficiency, the underlying system may choose to hold variable-size buffers for the file, but crucially, that buffering can be tuned to match available resources of memory, disc, and processor.

```
mcubeI :: Num a => CellEdge a -> Byte ->
                        XYZ -> [Triangle a]
mcubeI edges index xyz =
    group3 (map (selectEdge edges xyz)
                (mcCaseTable ! index))
```

This per-edge implementation guarantees that an interpolated vertices is computed only once and therefore no replication of the same value are present. When dealing with slow graphics hardware the possibility to reduce the amount of information sent to be rendered (i.e. duplicated primitives like vertices, edges or faces) is worth to be prosecuted.

Up to now we have built three different version of the same algorithm able to cope with the previously marked issues. Each version represents an independent and ready to execute program and at the same time as the flexibility to be merged with with its siblings to generate a unique optimized solution. We still miss one point worth of noting, it is well-known that in the original marching cubes, ambiguous cases can occur, and several efforts have been carried on in literature to enhance and generalize the original method to assure topological correctness of the result. Within the available solutions we adopted the approach proposed in [2, 13].

2.3 Functional Patterns

Occurrence of the same computational pattern are easily evinceable from the signature of each implemented function. The creation of a gluing interface for the suite of algorithm version is straightforward:

```
isosurface:: a -> Dataset a -> [Triangle b]
mcube:: a -> Cell a -> Format -> Dataset a
                            -> [Triangle b]
```

We pushed the abstraction a step further generalizing the interface to other surface fitting techniques in the means of marching tetrahedra and contour tracking. Both algorithm fall within the aforementioned specification. The marching tetrahedra algorithm is closely related to marching cubes except that the fundamental sampling structure is a tetrahedron instead of a cube. Extension of the marching cubes code presented so far, to the marching tetrahedra technique keeps unchanged the isosurface signature while the marching tetrahedra itself looks as follow.

```
mtetra :: (Num a, Int a, Fract b) => a ->
          TetraGrid a -> Cell a ->
          XYZ ->
          [ Triangle b ]
mtetra thresh g cell lookup =
  group3 map interpolate g ((!)mtCaseTable
              (caseIndex lookup cell thr))
```

The interpolation function remains unchanged as well while `mcCaseTable` is substituted with `mtCaseTable` table containing the possible configuration of the interpolant within a tetrahedral cell. Contour following defines a class of algorithm that given a cell intersected by the contour (isosurface in 3D) follows it walking along its neighbours until the starting point (the surface border) is reached. In our Haskell implementation we have split the contour following algorithm into two main functions: (1) Traverse Seeds: which given a Seed Set and a threshold value, searches the seeds sets for all the cells that constitute a seed for the given value. (2) Grow Contour: which given a Seed grows the contour following the contour path through cells adjacent to the seed. The function signature are expressed as follows:

```
traverseSeeds:: Dataset a -> a -> [Seed a]
                                -> [Triangle b]

growContour:: Dataset a -> a -> Seed a ->
                                  [Triangle b]
```

Our implementation of the Contour Tracking algorithm is built on top of the marching cube implementation, first the `Seed` data is introduced to define an arbitrary type of cell (in the regular case either a cube or a tetra) while `growContour` employs at its interior alternatively `mcube` or `mtetra` to extract the complete isosurface according to the seed (cell) kind.

2.4 Observations.

The approach presented so far has several interesting aspects. The availability of a suite of polymorphic versions of the same algorithm allows to choose between the more suitable ones for the type of resources available. Figures from Tables 3 and 4 show how performance can change between platforms with similar computational power but different architectures. Moreover through *polymorphic* types (sec 2.1) functions can be defined independent of the datatypes available on a specific architecture; type *predicates* allow developers to set out the minimum requirements that particular types must satisfy. Beyond the scope of this paper, *polytypism* [9], also known as structural polymorphism on types, has the capability to abstract over data organisation and traversal, e.g. a polytypic marching cubes could be applied to other kinds of dataset organization like irregular grids. The employment of a functional language like Haskell makes the construction of such a suite easy to achieve. Its abstraction power outlines common recursive patterns quite easily. If we consider the class of surface fitting techniques presented the emerging pattern is the one made up of a set of Cells (the Dataset) a threshold value and a fitting technique. The fitting technique itself (marching cubes, marching tetrahedra, trilinear interpolation) appears strictly dependent on the `Cell` kind (cube or tetrahedra) up to a generic and comprehensive definition that culminates in the `Seed` definition. The generic pattern can be then specified according to the kind of Dataset (Cell) and within each implementation further qualified with respect to specific computational issues. At the same time the intrinsic property of the language where each function is independent of the other ease the process of merging pieces

Table 1: Dataset Statistics.

dataset	size	wndw (b)	surface
neghip	64×64×64	4,16	131,634
hAtom	128×128×128	16,51	134,952
statueLeg	341×341×93	116,62	553,554
aneurism	256×256×256	65,79	1,098,582
skull	256×256×256	65,79	18,415,053
stent8	512×512×174	262,65	8,082,312
vertebra8	512×512×512	262,65	197,497,908

Table 3: Time Performance - Intel

	time (s)		
dataset	array	stream.	VTK
neghip	1.09	0.44	0.06
hydrogen	16.7	3.47	0.21
statueLeg	275.0	17.9	1.09
aneurism	619.7	28.1	1.73
skull	626.4	30.1	28.6
stent8	4530.0	79.7	13.1
vertebra8	6530.0	277.8	269.2

of code coming from different implementations but defined by logically equivalent signatures. In our testing phase for example we have noticed how the best performance on both machine were achieved by a marching cubes implementation which included both the streaming and sharing of the threshold computation.

2.5 Time and Space Profiles

Performance numbers are given for all the presented versions of marching cubes written in Haskell, over a range of sizes of dataset (all taken from `volvis.org`). The relevant characteristics of the datasets are summarised in Table 1, where the streaming window size is calculated as one plane+line+1. Tables 3 and 4 give the absolute time taken to compute an isosurface at threshold value 20 for every dataset, on two different platforms, a 3.0GHz Dell Xeon and a 2.3GHz Macintosh G5 respectively, compiled with the ghc compiler and `-O2` optimization. Table 2 shows the peak live memory usage of each version of the algorithm, as determined by heap profiling.

Table 2: Memory Usage

	memory (MB)		
dataset	array	stream.	VTK
neghip	0.270	0.142	1.4
hydrogen	2.10	0.550	3.0
statueLeg	11.0	3.72	15.9
aneurism	17.0	2.10	28.1
skull	17.0	2.13	185.3
stent8	46.0	8.35	119.1
vertebra8	137.0	8.35	1,300.9

On the Intel platform the array-based version struggles to maintain acceptable performance as the size of the array gets larger. We suspect that the problem is memory management costs: (1) The data array itself is large: its plain unoptimised representation in Haskell uses pointers, and so it is already of the order of 5–9× larger than the original file alone, depending on machine architecture. (2) The garbage collection strategy used by the compiler's runtime system means that there is a switch from copying GC to compacting GC at larger sizes, which changes the cost model. (3) The compiler's memory allocator in general uses block-sizes up to a maximum of a couple of megabytes. A single array value that spans multiple blocks will impose extra administrative burden. In contrast, the time performance of the streaming version scales linearly with the size of dataset outperforming VTK (see Table 4). It can also be seen that the memory performance is exactly proportional to the size of the sliding window (plane+line+1). The streaming version has memory overheads too, mainly in storing and retrieving intermediate (possibly unevaluated) structures. However, the ability to stream datasets in a natural fashion makes this approach much more scalable to large problem sets. Comparing the haskell implementations with implementations in other languages something interesting appears.

Table 4: Time Performance - PowerPC

	time (s)		
dataset	array	stream.	VTK
neghip	1.088	0.852	0.29
hydrogen	8.638	6.694	0.51
statueLeg	48.78	34.54	2.78
aneurism	72.98	54.44	5.69
skull	79.50	57.19	79.03
stent8	287.5	154.9	33.17
vertebra8	703.0	517.1	755.0

If we consider the figures obtained by running the algorithm on a Mac paltforms the streaming Haskell version is actually faster than VTK for the larger surfaces generated by skull and vertebra8. Moving to a different platform the panorama changes "apparently" in favour of the VTK implementation. However surprisingly the Haskell implementation appears to be still competitive. The interesting aspect worth noting is the speed-up gained by VTK which is much bigger than the one of the Haskell implementation, in proportion to the increase in computational power due to a faster processor. We suspect that such a difference resides in the compiler-generated code which could easily be better optimized for an Intel architecture rather than for a PowerPC.

3 Related Work

The difficulties of working with large volumes of data have prompted a number of researchers to consider whether approaches based on lazy or eager evaluation strategies would be most appropriate. While call-by-need is implicit in lazy functional languages, several

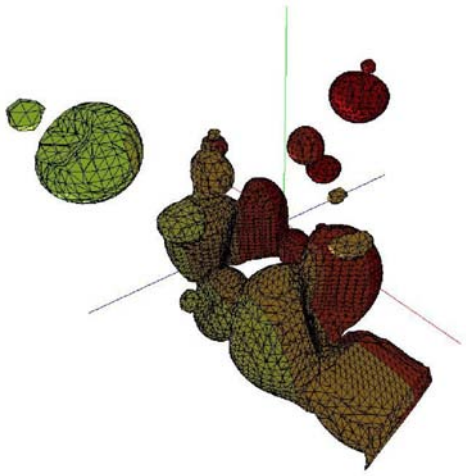


Figure 3: Functionally surfaced dataset coloured to show age of triangles in stream generated by the streaming marching cubes Haskell implementation over the neghip dataset.

efforts have explored more *ad hoc* provision of lazy evaluation in imperative implementations of visualization systems e.g. [11],[4]. In [15] Moran et al. use a coarse-grained form of lazy evaluation for working with large time-series datasets. The fine-grained approach inherent in Haskell not only delays evaluation until it is needed, but also evaluates objects piecewise. This behaviour is of particular interest in graphics and computational geometry, where order of declaration and computation may differ. Properties of our fine-grained streaming approach also match requirements for data streaming from [12], we refer for a more detailed disscussion on this topic to [5].

4 Conclusion

Our purely functional reconstruction of the marching cubes algorithm makes two important contributions. First, it shows how functional abstractions and data representations can be used to factor algorithms in new ways, in this case by replacing monolithic arrays with a stream-based window, and composing the overall algorithm itself from a collection of (functional) pipelines. This is important in the context of grid computing, because a stream- based approach may be more suitable for distribution than one that relies on a monolithic structure. It is also important for visualization, as streaming occupies an important niche between fully in-core and fully out-of-core methods, and the functional approach is novel in that the flow of data is managed on a need-driven basis, without the programmer resorting to explicit control over buffering. Second, the functional reconstruction shows that elegance and abstraction need not be sacrificed to improve performance; the functional implementation is polymorphic in the kind of data defining the sample points, and several of the simple functions used to make the final application are eminently reusable in other visualization algorithms (for example `mkStream`). Our next step is to explore how type-based abstraction, e.g. polytypic programming, can be used to make the algorithm independent of the specific mesh organization; we would like the one expression of marching cubes to apply both to regular and irregular meshes.

Acknowledgement

The work reported in this paper was funded by the UK Engineering and Physical Sciences Research Council.

References

[1] G. Allen, T. Dramlitsch, I. Foster, N.T. Karonis, M. Ripeanu, E. Seidel, and B. Toonen. Supporting efficient execution in heterogeneous distributed computing environments with cactus and globus. In *Supercomputing '01: Proc. of the 2001 ACM/IEEE conference on Supercomputing*, pages 52–52, 2001.

[2] E. Chernyaev. Marching cubes 33: Construction of topologically correct isosurfaces. Technical report, 1995.

[3] D. Clarke, R. Hinze, J. Jeuring, A. oh, and J. de Wit. The generic haskell user's guide, 2001.

[4] M. Cox and D. Ellsworth. Application-controlled demand paging for out-of-core visualization. In *Proceedings of Visualization '97*, pages 235–ff. IEEE Computer Society Press, 1997.

[5] D. Duke, M. Wallace, R. Borgo, and C. Runciman. Fine-grained visualization pipelines and lazy functional languages. *IEEE Transaction on Visualization and Computer Graphics*, 12(5), 2006.

[6] R.B. Haber and D. McNabb. Visualization idioms: A conceptual model for scientific visualization systems. In *Visualization in Scientific Computing*. IEEE Computer Society Press, 1990.

[7] Haskell: A purely functional language. http://www.haskell.org, Last visited 27-03-2006.

[8] J. Hughes. Why functional programming matters. *Computer Journal*, 32(2):98–107, 1989. See also http://www.cs.chalmers.se/ rjmh/Papers/whyfp.html.

[9] Johan Jeuring and Patrik Jansson. Polytypic programming. In J. Launchbury, E. Meijer, and T. Sheard, editors, *Tutorial Text 2nd Int. School on Advanced Functional Programming, Olympia, WA, USA, 26–30 Aug 1996*, volume 1129, pages 68–114. Springer-Verlag, 1996.

[10] R. Lämmel and S. Peyton Jones. Scrap your boilerplate: a practical design pattern for generic programming. *ACM SIGPLAN Notices*, 38(3):26–37, 2003. Proceedings of the ACM SIGPLAN Workshop on Types in Language Design and Implementation (TLDI 2003).

[11] D.A. Lane. UFAT: a particle tracer for time-dependent flow fields. In *Proceedings of Visualization '94*, pages 257–264. IEEE Computer Society Press, 1994.

[12] C.C. Law, W.J. Schroeder, K.M. Martin, and J. Temkin. A multi-threaded streaming pipeline architecture for large structured data sets. In *Proceedings of Visualization '99*, pages 225–232. IEEE Computer Society Press, 1999.

[13] T. Lewiner, H. Lopes, A.W. Vieira, and G. Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003.

[14] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of SIGGRAPH'87*, pages 163–169. ACM Press, 1987.

[15] P.J. Moran and C. Henze. Large field visualization with demand-driven calculation. In *Proceedings of Visualization'99*, pages 27–33. IEEE Computer Society Press, 1999.

[16] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Prentice Hall, second edition, 1998.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Application of Fault Injection to Globus Grid Middleware

Nik Looker [(a)], Jie Xu [(a)], Tianyu Wo [(b)], Jinpeng Huai [(b)]

[(a)] University of Leeds, Leeds. LS2 9JT, UK
[(b)] Beihang University, Beijing 100083, PRC

Abstract

Dependability is a key factor in any software system and has been made a core aim of the Globus based China Research Environment over Wide-area Network (CROWN) middleware. Our past research, based around our Fault Injection Technology (FIT) framework, has demonstrated that Network Level Fault Injection can be a valuable tool in assessing the dependability of RPC oriented SOAP based middleware. We present our initial results on applying our Grid-FIT method and tools to Globus middleware and compare our results against those obtained in our previously published work on Web Services. Finally this paper outlines our future plans for applying Grid-FIT to CROWN and thus providing dependability metrics for comparison against other middleware products.

1 Introduction

The Globus Toolkit [1] is an open source software toolkit used for building Grid systems and applications [2]. Since it is the front running technology for Grid computing its dependability is a key issue. A large part of the infrastructure for Globus Toolkit 4 is constructed around Java Web Services utilizing Apache Axis [3] as the transport and Apache Tomcat [4] as the container.

Fault Injection is a well-proven method for assessing the dependability of a system. Although much work has been done in the area of fault injection and distributed systems in general, there appears to have been little research carried out on applying this to Web Service based systems [5, 6]. Network Level Fault Injection has provided promising results in assessing the dependability of SOA [7].

Web Service – Fault Injection Technology (WS-FIT) [7] is a dependability assessment method and tool for assessing Web Services by fault injection. WS-FIT is targeted towards Java Web Services implemented using Apache Axis transport. Given the similarities in underlying middleware technology our aim has been to implement a second tool, Grid-FIT, targeted at Globus Grid Services.

This paper details the fundamental concepts behind our method and outlines the differences between WS-FIT and Grid-FIT. We demonstrates Grid-FIT with a number of proof of concept experiments and compare the results against similar results obtained from WS-FIT. Finally we outline our future plans to apply Grid-FIT to a middleware product implemented over Globus Toolkit to provide data that will allow us to construct a fault model and failure modes to assess Grid middleware and also provide dependability metrics for the middleware products.

2 Service Technology

Globus is built upon Web Service middleware and FIT manipulates messages exchanged between Web Services to assess service dependability.

A Web Service is a software service defined by a number of standards that can be used to provide interoperable data exchange and processing between dissimilar machines and architectures. For the purposes of our research we are concerned with Web Services defined by the W3C that are described by WSDL [8] and implemented using SOAP [9] and the RPC model described in that document.

2.1 WSDL

Web Services Description Language (WSDL) is an XML-based Interface Definition Language (IDL) used to define Web Services and how to access them [8, 10]. Our research is mainly concerned with RPC message exchanges. WSDL lends itself well to providing explicit information on the structure of message exchanges between Web Services and their clients. WSDL documents are made up of a number of XML elements and this gives us a good starting point for producing fault injection triggers.

Table 1 shows an example wsdl:message. A wsdl:message is composed of an element that has a unique name attribute that is used to identify the message and a number of wsdl:part

elements. Each wsdl:part defines a parameter (or return value in the case of a response message). A wsdl:part has an associated name that must be unique within the wsdl:message element and a type that defines the parameter type. There are a number of predefined types and complex types can also be defined using a types element.

```
<wsdl:message
  name="unregisterServiceRequest">
  <wsdl:part
    name="context" type="xsd:string"/>
  <wsdl:part
    name="entry"
   type="impl:ServiceEntry"/>
</wsdl:message>
```

Table 1: Example WSDL Message

Once all request and response messages required to implement an RPC style interface have been defined they can be used to define the calling interface for the Web Service. This is done by use of the wsdl:portType element (see Table 2). A wsdl:portType contains a number of wsdl:operation elements with each wsdl:operation element corresponding to a method of the Web Service. Each wsdl:operation is made up of a wsdl:input element that will be the request part of the RPC and a wsdl:output element that will be the response message of the RPC.

```
<wsdl:portType name="Quote">
  <wsdl:operation name="getQuote"
    parameterOrder="symbol">
    <wsdl:input
      name="getQuoteRequest"
      message="impl:getQuoteRequest"/>
    <wsdl:output
      name="getQuoteResponse"
      message="impl:getQuoteResponse"/>
  </wsdl:operation>
  <wsdl:operation
name="unregisterService"
     parameterOrder="context entry">
    <wsdl:input
     name="unregisterServiceRequest"

message="impl:unregisterServiceRequest"/
>
    <wsdl:output
     name="unregisterServiceResponse"
     message=

"impl:unregisterServiceResponse"/>
  </wsdl:operation>
</wsdl:portType>
```

Table 2: Example WSDL PortType

The above explanation briefly describes the use of WSDL to define a classic RPC style Web Service. WSDL can also be used describe other styles of Web Service calling interface such as message oriented calling but this is outside the scope of this research.

2.2 SOAP

SOAP [9, 10] is a messaging protocol designed to allow the exchange of messages over a network. It is XML based to allow the exchange of messages between dissimilar machines.

Our method is primarily concerned with RPC mechanisms over SOAP. This is defined by the W3C in [11] and describes a general purpose RPC mechanism. The message types that are involved in an RPC exchange and the relevant features used by our method are briefly reviewed here.

A SOAP message utilizes the http://schemas.xmlsoap.org/soap/envelope/ schema which defines the namespace soapenv and this namespace is setup in the soapenv:Envelope element. Consequently all elements that unitize this namespace must be enclosed by the soapenv:Envelope element. The soapenv namespace defines a semantic framework for SOAP messages.

The body of the SOAP message is enclosed by the soapenv:Body element. This element acts as a grouping for the body elements for different types of messages. Its primary function is to keep the body elements distinct from other grouping of elements, for instance a header block.

These two elements form the basis of a SOAP message. The soapenv:Body element is then populated with elements that make up the payload of a request, response or fault message.

A typical request message is given in Table 3. The request message name is defined in the wsdl:operation (see Table 2) but by convention the name of the message equates to the service method name but it can be defined as any valid name. In the example the message and method name are getQuote. The message element is therefore ns1:getQuote. The namespace ns1 is defined to be the urn of the service that implements the method. If this is combined with the address of the server hosting the service this allows a specific method of a specific service on a specific server to be identified.

The ns1:getQuote element contains parameter elements that represent the RPC parameters, for instance the getQuote method takes one string parameter called symbol so ns1:getQuote contains one element with an element tag symbol which contains the string data for that parameter. Parameters are defined in WSDL by wsdl:part elements in wsdl:message elements (see Table 1).

A typical response message is given in Table 4. The response message is similar in structure to the request message but by convention the response message name is post-fixed with the word Response although, again, it can be any valid name defined in the wsdl:operation element. In this example the response element name is ns1:getQuoteResponse.

```
<soapenv:Envelope
 xmlns:soapenv=

"http://schemas.xmlsoap.org/soap/envelope
/"
 xmlns:xsd=
  "http://www.w3.org/2001/XMLSchema"
 xmlns:xsi=
  "http://www.w3.org/2001/XMLSchema-
instance">
 <soapenv:Body>
  <ns1:getQuote
   soapenv:encodingStyle=
"http://schemas.xmlsoap.org/soap/encoding
/"
    xmlns:ns1=
"http://quote.stock.samples.dasbs.org">
     <symbol
      xsi:type="xsd:string">
      foo
     </symbol>
   </ns1:getQuote>
  </soapenv:Body>
</soapenv:Envelope>
```

Table 3: Typical Request Message

A response element contains elements that represent any method return value and any parameters that are marked to be marshalled in-out or out. Method return results follow the naming convention of the method name post fixed by the word Return and are represented in the WSDL by wsdl:part elements. In-out and out parameters follow the same conventions as parameters in a request message.

The example response message in Table 4 also demonstrates the format that objects and arrays are marshalled in a SOAP message. This utilizes the multiRef element. Each object or item in an array is created using a multiRef that has an id. The actual parameter/return value is then set to this reference id and the complex data can be constructed within the multiRef element in the same way that individual parameters are constructed in a message. This technique applies to both request and response messages.

Table 5 shows a typical SOAP Fault Message. Fault Messages are used to return failure information from a server to a client. The soapenv:Fault element contains three elements: faultcode, faultstring and detail. These elements are used to convey failure information to the client with the faultstring and detail elements being language specific, for instance using Axis 1.1 for Java if a piece of user code on a server throws a Java exception the faultcode is set to soapenv:Server.userException to indicate that the fault originates in server side user code, then the fault string is set to the text description of exception and the detail element is not used.

```
<soapenv:Envelope
    xmlns:soapenv=…
    xmlns:xsd=…
    xmlns:xsi=…>
 <soapenv:Body>
    <ns1:getQuoteResponse
      soapenv:encodingStyle=…
      xmlns:ns1=…>
      <getQuoteReturn href="#id0"/>
    </ns1:getQuoteResponse>
    <multiRef id="id0"
      soapenc:root="0"
      soapenv:encodingStyle=…
      xsi:type="ns2:QuoteValue"
      xmlns:soapenc=…
      xmlns:ns2=
     "http://quote.stock.samples.dasbs.org">
        <date
          xsi:type="xsd:dateTime">
           2004 10 30T10:54:18.511Z
        </date>
        <quote
          xsi:type="xsd:double">
           47.5
        </quote>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

Table 4: Typical Response Message

```
<soapenv:Envelope xmlns:soapenv=…
                  xmlns:xsd=…
                  xmlns:xsi=…
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
      <faultstring>
        java.rmi.RemoteException:
        can&apos;t get a stock price
      </faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Table 5: Typical Fault Message

3 Grid Middleware

Grid middleware allows the construction of Grid services. It typically provides a transport mechanism as well as security facilities that give a consistent set of operations over a heterogeneous platform base.

Globus Toolkit is the front running reference implementation of OGSA. Version 4 of this product is based around a Java core composed

of Apache Tomcat and Apache Axis that makes the Globus environment similar to our previous experimental environments.

The major difference is the way that Globus packages and exchanges data. Whilst Globus utilizes the standard Axis RPC mechanism it packages parameters and return values into a complex data structure. This means that each message exchange has a schema associated with it (See Table 6) rather than each parameter being defined as a set of wsdl:part in the wsdl:message element.

```
<xsd:element name="getQuote">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element
          name="symbol"
          type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
```

Table 6: Example Globus WSDL

This means that the WSDL for each request and response message has a more nested structure but this has no major effect on the structure of the messages exchanged. This is because when there is only one object passed as a parameter it can be contained in the message body rather than as a sequence of multiRef elements (See Table 7).

```
<soapenv:Body>
  <getQuote xmlns="http://…">
    <symbol>foo</symbol>
  </getQuote>
</soapenv:Body>
```

Table 7: Example Globus SOAP Message

Whilst some modification to the code base is necessary to accommodate these differences in WSDL they are fairly minor.

3.1 CROWN

To fully evaluate our system we require a complex application to access. In doing so we would gain useful knowledge on how to construct fault models and where to apply them in a system. With this in mind we have selected a test bed system which will provide us with a real-world test senario.

CROWN is a Grid test bed to facilitate scientific activities in different disciplines. The CROWN middleware platform is consist of 3 parts. Firstly, many resources (including computers, clusters and some storage devices) should be connected via a nation-wide network infrastructure to build the CROWN architecture. Secondly, a series of grid middleware and auxiliary tools should be provided to meet the common requirements of different scientific activities in all kinds of areas. Finally, a number of typical applications should be put in use over the fabric and middleware to demonstrate the feasibility and robustness of CROWN.

The CROWN NodeServer is the basic environment for Grid services to be deployed and executed in the CROWN system. It has 5 main features that are outlined here.

Grid service container: Based on the Globus Toolkit WS Core 4.0, CROWN NodeServer implements a fundamental Grid service hosting environment that follows the OGSA/WSRF specifications.

Remote and hot service deploying: The CROWN NodeServer allows service providers to deploy and configure Grid services remotely without having to restart the container. This is combined with a security and trustworthy mechanism to provide a way for distributing applications through the CROWN system to execute securely and enhances the scalability of the whole system.

Resource monitor: CROWN NodeServer gathers both static and dynamic information of the underlying resource, and shares them via a Grid service interface. The information can be used to control workflow or monitor a system.

Legacy applications integration: Many applications are not developed in a service oriented way. To add them into Grid environment, CROWN NodeServer can encapsulate these programs as Grid services.

Security and trust: The NodeServer follows the WS-Security specification and provides security functions of SOAP signature and encryption, authentication, authorization and identity mapping between different security infrastructures. By using automated trust negotiation, NodeServer can help to setup trust relationship among strangers.

By using NodeServer, we encapsulated heterogeneous resource into a single homogeneous view. All kinds of resources in CROWN are provided via the services running in NodeServer.

We have selected the NodeServer as our first real-word Grid-FIT test case because it gives us a complex system to test, is written utilizing SOAP and it's assessment would be a benefit to both parties involved in the collaboration, i.e. The middleware can be made more reliable and Grid-FIT can use the data gathered to construct better fault models.

4 Grid-FIT Method

Grid – Fault Injection Technology (Grid-FIT) [6, 7, 12, 13] is a tool derived from our WS-FIT tool and is a fault injector that allows network level fault injection to be used to assess Grid middleware. Grid-FIT has been implemented specifically to assess Globus systems. The implementation specifically handles the problems associated with modifying messages when signing and encryption are being used [14].

Perturbation [15] attempts to forcefully modify program states without mutating existing code statements. This is often achieved by code insertion. Instrumented code (termed perturbation functions) is added to a system in the form of function calls that modify internal program values. These modified values can either be generated based on the original value, generated randomly or a fixed constant value can be used. This technique is useful in testing such things as fault tolerance mechanisms. Our method extends this method to make meaningful perturbation to a SOAP message, e.g. our method can target a single parameter within an RPC message sequence. Since Grid-FIT can effectively achieve this without the need for modification or access to the Service source code it is less invasive that code insertion [16].

Our method uses an instrumented SOAP API that includes two small pieces of hook code. One hook intercepts outgoing messages, transmits them via a socket to the fault injector engine and receives a modified message from the fault injector. This message is then transmitted normally to the original destination. There is a similar hook for incoming messages, which can be processed in the same way (See Figure 1).

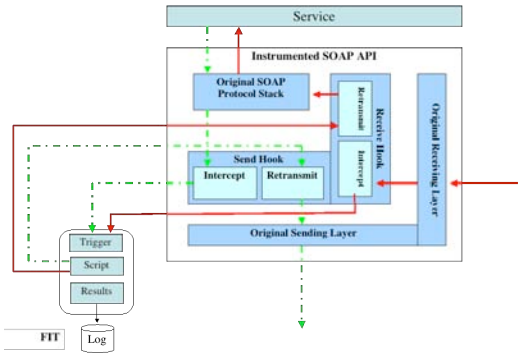


Figure 1: Message Interception and Injection

Our triggering mechanism is based around specific messages being received. These messages can further be decomposed to specific parameters within a specific message so a particular fault model can be applied to a specific parameter.

Triggers are constructed based on the WSDL for a Web Service (see Section 2.1). The WSDL definition for a Web Service contains detailed information on the messages exchanged. The GUI can import WSDL and parse the wsdl:message, wsdl:portType and wsdl:operation elements into a taxonomy that can be used to construct triggers. By mapping the wsdl:message elements onto soapenv:body elements in a message stream and further mapping wsdl:part elements onto RPC parameters contained in a soapenv:body we can construct precise triggers that allow perturbation of individual parameters. This indicates the required part of a SOAP message and a fault can be injected at this point such as perturbing a parameter.

5 Case Study

The purpose of this case study is to demonstrate that Grid-FIT operates in a similar fashion to the WS-FIT method, whilst not causing any unintentional degradation in the operation of the Globus middleware. To achieve this we have adapted a previously experiment based around WS-FIT and Web Services so that the results can be compared [17].

In Looker et al [17] these tests were constructed using the application of the Extended Fault Model (EFM) implemented by FIT but due to time constraints the results given here were obtained by manually adapting the test scripts rather than reapplying the EFM.

5.1 Test Scenario

To provide a test bed to demonstrate Grid-FIT we have constructed a test system that simulates a typical stock market trading system. This system is composed of a number of elements: 1) A service to supply real-time stock quotes; 2) A service to automatically trade shares; 3) A bank service that provides a simple interface to allow deposits, withdrawals and balance requests; 4) A client to interact with the SOA (See Figure 2).

In our original experiment the services were implemented as Axis 1.1 Web Services running under Tomcat 5.0.28. Our new experiment implements these services as Globus 4.0.1 services although they utilize the same algorithms as before.

We have implemented our stock quoting service to use a large repeatable dataset, stored in a backend database to produce a time based real-time stock quote. Since the quote service is based around a database containing the simulated quote values it is possible to replicate a test run exactly by resetting time etc. to a set of starting conditions.

Our trading service implements a simple automatic buying and selling mechanism. An upper and lower limit is set which triggers trading in shares. Shares are sold when the high limit is exceeded and shares are brought when the quoted price is less than the lower limit.

The buying and selling process involves transferring money using the bank service and multiple quotes (one to trigger the transaction and one to calculate the cost). Since these multiple transactions involve processing time and network transfer time this constitutes a race condition as our quoting service produces timed real-time quotes. Any such race condition leaves the potential for the system to lose money since the initial quote price may be different from the final purchase price. This is intentionally to demonstrate that Grid-FIT does not introduce a significant overhead into the system and thus effect its operation.

This paper details three different series of data: 1) A baseline set of data with the system running normally; 2) A simulated faulty/malicious service 3) A simulated heavily loaded server.

Our test system was implemented using Globus WS-Core Version 4.0.1 running on Apple Mac OS X using G4 1.5 GHz processors and 1Gb RAM.

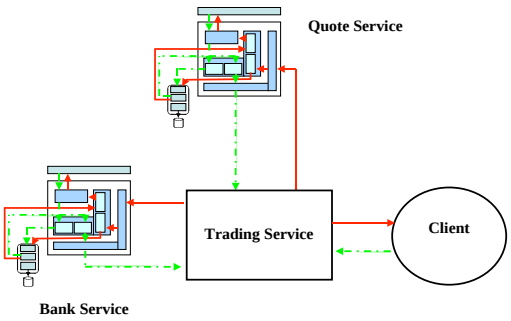


Figure 2: Instrumented System

5.2 Baseline

Our original baseline experiment using Web Services from the normally running system allowed us to verify that the system operates according to its specification. The system provided matches for transactions 99.8% averaged over a number of runs. The average RPC time was measured at 0.05s and appeared to allow the algorithms to function correctly.

The test case was iterated and the transactions from each were compared. Apart from minor timing variations the analysis showed that the test case was repeatable when Web Service technology was used.

We repeated this experiment with our instrumented Globus system (See Table 8). This gave results similar to the Web Service based system with an average match of 99.5% and an average RPC execution time of 0.05s.

	Test 1		Test 2		Test 3		Test 4	
	Match	Mismatch	Match	Mismatch	Match	Missmatch	Match	Missmatch
Match %	100.00	0.00	99.00	1.00	100.00	0.00	99.00	1.00
Average Time	0.05		0.05	0.07	0.05		0.05	0.06
Std Dev	0.02		0.02		0.03		0.02	

Average Match	99.50
Std Dev	0.57735

Table 8: Baseline Data For Globus System

By comparing the test runs there appears to be very little difference between the data sets obtained except for minor timing variations (See Figure 3).

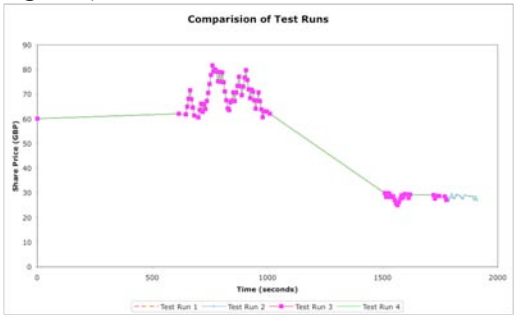


Figure 3: Comparison of Baseline Test Data Runs

By comparing the previous Web Service based data with the new Grid-FIT data we observed no significant differences.

5.3 Fault/Malicious Service

The second test series simulated a faulty/malicious quote service by applying a random fault model. The random model used injects a normally distributed randomly generated value that replaces the RPC parameter specified. The random sequence was hard coded into the script to allow repeatability of the test. We used the same starting conditions as the first test series and iterated the test series four times.

For the Web Service based system all transactions produced a mismatch as expected, since each quote value would be corrupted. The Globus based system was then used to run the experiment. This produced a similar result with all transactions being mismatch transactions (See Table 9).

	Test 1		Test 2		Test 3		Test 4	
	Match	Mismatch	Match	Mismatch	Match	Missmatch	Match	Missmatch
Match %	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00
Average Time		0.04		0.04		0.04		0.04
Std Dev		0.00		0.00		0.00		0.00

Average Match	0.00
Std Dev	0

Table 9: Globus Attack Data

By comparing the data series against each other we can see that the experiment is repeatable (See Figure 4).

Figure 4: Comparison of Attack Test Data Runs

Again by comparing the Web Service based data with the data obtained in this test case we observed no significant differences.

5.4 Latency Injection

The final series of data in this set of tests again injected a fault into the system. This fault was an increased latency induced into the quote service. This latency simulates server loading. To implement this we introduced a delay into the system based on a poisson distribution. The distribution is statically encoded into the test script to allow for repeatability. The test was iterated over four runs.

	Test 1		Test 2		Test 3		Test 4	
	Match	Mismatch	Match	Mismatch	Match	Missmatch	Match	Missmatch
Match %	61.00	39.00	61.00	39.00	60.00	40.00	59.00	41.00
Average Time	2.90	6.96	2.90	6.96	2.95	6.79	2.71	6.92
Std Dev	3.27	2.02	3.27	2.02	3.28	2.25	3.26	1.98

Average Match	60.25
Std Dev	0.95743

Table 10: Globus Latency Data

Table 10 contains the results from the injection performed on the quote service. This clearly indicates that the system is functioning differently to the baseline test series. By analysis the test data gathered we could see that the quote value that triggers a sale/purchase of shares differs from sale/purchase price approximately 40% of the time. This is due to some quote values being delayed long enough to cause the quote to fall into the next quote period.

This is significantly smaller that our Web Service experiment which gave a result of 63%. Whilst a detailed investigation of this result is required this could be due to variations in the statically encoded distribution and minor differences in timings introduced by Globus. Small timing variations introduced into this algorithm could produce large effects since each variation would alter a share trading event and influence the amount of money available for the next share trade.

We compared the test runs and the results are given in Figure 5. This shows that whilst the individual test runs vary more that the previous two sets of data they follow the same trend and are repeatable. This is comparable with the experimental data gathered from the Web Service based system.

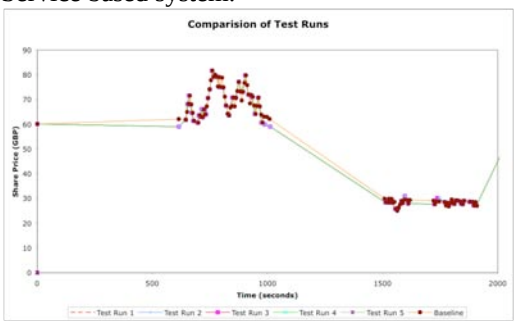


Figure 5: Globus Latency Comparison

6 Conclusion and Future Work

This paper has detailed the application of our FIT based tool, Grid-FIT, to Globus version 4. We have demonstrated that our dependability method previously developed for Web Service based systems can be applied with little modification to Grid based systems.

We have demonstrated that the differences in definition and message format are negligible with regard to our tool. A minimum of work was required to implement our new Grid-FIT tool from the code base designed for WS-FIT.

Our experimental data from this paper demonstrates that a Grid based system can be assessed using the same method as has been applied in previous research and produces similar results with no apparent unintended impact on the Globus middleware except during the latency injection.

We observed some differences when latency injection was performed between the Web Service based test case and the Globus based test case. This test case is the most sensitive of all tests to timing constraints so a possible cause may be that slightly different timing constrains cause by Grid-FIT and Globus Toolkit caused slightly different control pathways to be executed at different times. Even so this test

case followed the same basic trend as the Web Service based tests.

This work provides us with a proof of concept. As a next step we intend to implement an enhanced fault and failure model to facilitate typical Grid system assessment based on the framework provided by FIT.

To accomplish this we will need experimental data on a non-trivial Globus based system.

We intend to utilize the CROWN middleware to provide us with a test bed system. By applying Grid-FIT to the NodeServer element built into CROWN we will gain valuable experience and data on assessing Globus based systems. This collaboration will be of advantage to both sides since dependability is a key aim of the CROWN middleware and by analyzing the application of our Grid-FIT tool and method to CROWN we will gain valuable data on how to enhance our fault models.

7 References

[1] I. Foster, "Globus Toolkit Version 4: Software for Service-Oriented Systems," presented at IFIP International Conference on Network and Parallel Computing, 2005.

[2] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, and J. V. Reich, "The Open Grid Services Architecture, Version 1.0.," Global Grid Forum (GGF) 2005.

[3] Apache, "Axis Architecture Guide," vol. 2003, 1.1 ed: Apache Axis Group, 2003.

[4] "The Apache Jakarta Tomcat 5 Servlet/JSP Container," 2006.

[5] E. Marsden, J. Fabre, and J. Arlat, "Dependability of CORBA Systems: Service Characterization by Fault Injection," presented at Symposium on reliable distributed systems, Osaka, Japan, 2002.

[6] N. Looker and J. Xu, "Assessing the Dependability of SOAP-RPC-Based Web Services by Fault Injection," *9th IEEE International Workshop on Object-oriented Real-time Dependable Systems*, pp. 163-170, 2003.

[7] N. Looker, M. Munro, and J. Xu, "A Method for Dependability Analysis of Web Services," *Information and Software Technology, Elsevier*, 2006.

[8] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL)," 1.1 ed: W3C, 2001.

[9] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer, "Simple Object Access Protocol (SOAP) 1.1," 1.1 ed: W3C, 2000.

[10] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, "Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI," *IEEE Internet Computing*, vol. 6, pp. 86-93, 2002.

[11] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, and H. F. Nielsen, "SOAP Version 1.2 Part 2: Adjuncts," vol. 2005: W3C, 2003.

[12] N. Looker, M. Munro, and J. Xu, "WS-FIT: A Tool for Dependability Analysis of Web Services," presented at The 1st Workshop on Quality Assurance and Testing of Web-Based Applications, COMPSAC, Hong Kong, 2004.

[13] N. Looker, M. Munro, and J. Xu, "Simulating Errors in Web Services," *International Journal of Simulation Systems, Science & Technology*, vol. 5, 2004.

[14] N. Looker and J. Xu, "Assessing the Dependability of OGSA Middleware by Fault Injection," *Proceedings of the Symposium on Reliable Distributed Systems*, pp. 293-302, 2003.

[15] J. Voas, "Fault Injection for the Masses," *Computer*, vol. 30, pp. 129-130, 1997.

[16] N. Looker, M. Munro, and J. Xu, "A Comparison of Network Level Fault Injection with Code Insertion," presented at 29th Annual International Computer Software and Applications Conference (COMPSAC'05), Edinburgh, Scotland, 2005.

[17] N. Looker, B. Gwynne, J.Xu, and M. Munro, "An Ontology-Based Approach for Determining the Dependability of Service-Oriented Architectures," presented at 10th IEEE International Workshop on Object-oriented Real-time Dependable Systems, Sedona, USA, 2005.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Grid Single Sign-On in CCLRC

Jens Jensen and David Spence and Matthew Viljoen
CCLRC Rutherford Appleton Laboratory

Abstract

This paper presents the latest results in on-going work on developing a single sign-on solution to access Grid resources. Since last year's e-Science All Hands Meeting, we have adapted a Java terminal by integrating it with site authentication infrastructures to provide access to the NGS and CCLRC's SCARF cluster, using MyProxy to manage the certificates and proxies that are essential for Grid access. We describe the architecture and details of the implementation, and how it fits into the site infrastructure, as well as future Shibboleth deployments. Although the work is done at CCLRC, this work is applicable to any site with a Kerberos or Active Directory infrastructure, and will be of interest to anyone working with authentication technologies.

1 Introduction

At last year's AHM we presented plans for implementing Single Sign-On (SSO) to Grid resources at CCLRC, including the NGS. We demonstrated a prototype portal with a simple job management workflow and web-based terminal access.

In this paper we describe our achievements and experiences in providing full terminal access. Thus, this paper focuses mainly on providing *terminal* access, as opposed to *portal* access. CCLRC has made no effort to further develop the portal prototype [1], nor to integrate SSO with existing Grid portals, but the terminal described in this paper is successfully in production. Grid access is integrated with wider site SSO efforts. We briefly describe how this work integrates with other SSO work, and related future directions.

1.1 Background

Many projects claim to have single sign-on. To these projects this means that a user only has to type the password or passphrase protecting their credentials once, or once every day. For example, Globus has the Globus proxy [2], which enables a user to access Globus resources without having to retype the passphrase that protects the user's (X.509) private key. `ssh` has the ssh "agent": a user space daemon which caches the unprotected (ssh) private key in memory.

On the other hand, for two primary customers of CCLRC's internal Grid resources (Diamond and ISIS), SSO means *integrated user management*. They want their user offices to be able to set up accounts for new users, including visiting scientists, and they want their databases to be consistent.

It is thus useful to give a definition of SSO, and we do this in section 1.2.

1.2 Aims and Use Cases

The principal use case for this work are:

1. A user on site logs into the site's Microsoft Active Directory (AD) [3] system – which, for the purposes of this paper, is compatible with Kerberos V [4]. Using this token, or ticket, the user is able to access the Grid.

2. Users who have certificates from the e-Science Certification Authority (CA), or from another Grid CA, can access the Grid resources using their certificates, both on site and off site. For users who have both local (AD) accounts and a certificate, the identity management knows that those two identities belong to the same person. In other words, the user can access the same resources with AD on site and using their personal certificate from a laptop when they're travelling.

3. Finally, for the user who does not have access either to the local account, or to their e-Science certificate, the terminal falls back to asking for username and password. The username and password are both the ones the user would use to authenticate locally.

The conditions were also that the software should be easy to install; for example, a user should be able to install and run it without having privileged access to their own desktop computers.

Finally, since many users were using `ssh` based login, we need to migrate them off that: they must get certificates, the certificates must grant them access to the same resources that they could access via `ssh`, and in particular, their identity must be preserved — i.e., we need to map between the `ssh` public key and the certificate public key.

It is clear that identity management is an essential part of SSO. Moreover, identity management is quite complicated, particularly over longer timescales. We will touch upon the subject briefly in the next section and in the Architecture section below, but otherwise it is outside the scope of this paper.

1.3 Integrating SSO

As mentioned above, the terminal work integrates with other site SSO efforts. The user offices of the CCLRC facilities will manage the user accounts for both local and external users in CCLRC's staff database. The challenge for the facilities is to populate the database with consistent data, because a given user's details may not be up to date, or the user may be present more than once in the database – for example, the user may have registered with more than one experiment.

For our purpose, we need to add the Grid identity information (mainly the Distinguished Name (DN)) to this database. Populating it initially is already a challenge, because 200 personal certificates have been issued to CCLRC and Diamond, and need to be matched to the Active Directory id (or Kerberos name). Strictly speaking, each user must do it for themselves by proving *both* their identities to the database – or, more precisely, an agent running on behalf of the user which is able to ask the user for proof of both identities (or pick up SSO versions of both), contact a server which can pick up both tokens and update the database.

1.4 Other Work

In this section we briefly describe other efforts in the area of SSO. The simplest example of SSO is of the site that is fully "Kerberised," where Kerberos authentication grants access to all resources. This doesn't immediately support roaming users, but it does provides SSO within the site. For Grid use such a site will have to implement *credential conversion* via MyProxy [5] or KCA (which provide credential conversion, generating short-lived certificates for users presenting Kerberos tickets). This is the approach taken by Fermilab. A KCA converts a Kerberos ticket to a short-lived X.509 certificate (similar to a Globus proxy), but these certificates cannot be used at many external sites because KCAs are not fully trusted compared to *classic* CAs [6] (i.e., CAs that issue long-lived end-entity certificates), although this is slowly improving. Another major disadvantage is that this approach does not scale: current Grid middleware needs *all* CAs in all hierarchies installed along with their signing policies and CRL endpoints. If each institution has its own CA the international collaborations would be overwhelmed by the sheer number of CAs. The same problem arises in the trust context: each CA has to negotiate trust with peer CAs and international resource providers, and they would become overwhelmed if there were more than one CA per country. One could argue that it should be sufficient to review the root's policy, but most resource providers need to know details of each individual site's credential conversion policy: who can get an account, how is the DN tracked back to the identity, etc.

Another approach was taken by Brookhaven National Laboratories. Like CCLRC, they were not keen on having disparate `ssh` and Grid key management infrastructures in addition to their site Kerberos infrastructure. They integrated a One Time Password (OTP) system with parts of their site infrastructure [7]; they found it was simple to integrate, but the cost was relatively high since users had to have hardware OTP tokens.

2 Architecture

Figure 1 gives an overview of the architecture of the CCLRC SSO solution. In the left half of Figure 1 the site authentication system is employed, which is the only authentication infrastructure users are concerned with. Depending on the implementation of SSO the site authentication token can be anything from a password to a Kerberos [4] ticket and correspondingly the user may (in the case of passwords) or may not (in the case of a Kerberos ticket) have to be actively involved in supplying the token. From the user's point of view, the difference is whether they have to type their password only once per session (e.g., when logging on to a computer, case 1 from Section 1.2), or whether they have to type it again when accessing the user interface (case 3) — but at least then, it is the *same* username and password as the federal one. In either case, in the architecture, the token is passed on to the MyProxy [8] server by the user interface, which then checks its validity with the site authentication infrastructure.

Case 2 from Section 1.2 is subtly different. From the user's point of view, for users who already have eScience certificates, they must upload an unencrypted proxy to the MyProxy server, typically once every week. This requires authentication, so the MyProxy upload tool must be integrated with SSO.

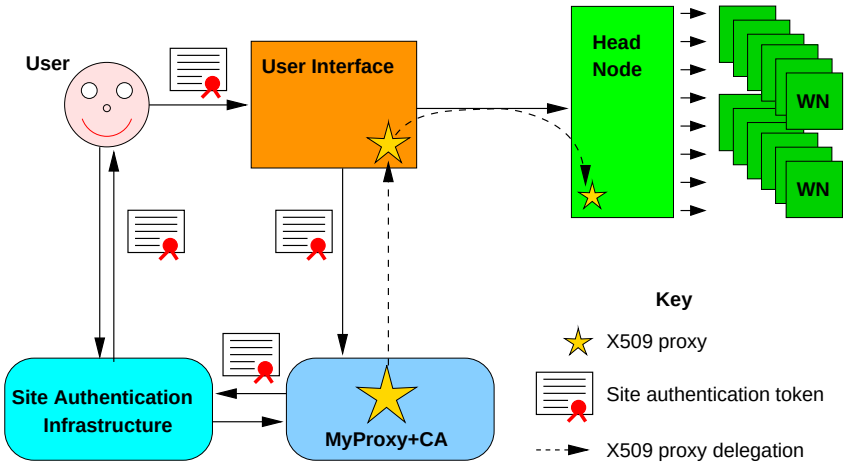


Figure 1: The architecture

They may have a passphrase for the browser's keystore and another one protecting the private key if the credentials are exported from the browser. These passphrases are independent of the SSO infrastructure — indeed, it is possible to leave the browser keystore or the exported key entirely unprotected (of course this violates the CA's policy).

In any case, users need to unlock their credentials, and to authenticate to the MyProxy server, using any method, to upload their proxy. Alternatively the user interface can directly use the local X.509 credentials (either in the browser, or exported) for authentication of the user. In this case the user will have to type the passphrase that unlocks the private key.

Grid access always requires an X.509 credential. For users without UK eScience certificates, we solve the SSO to the Grid problem by generating short-lived, low-assurance (see below) credentials within the MyProxy server.

The right of Figure 1 is within the Grid Security Infrastructure (GSI) domain and uses X.509 certificates [9] and proxies. Within the SSO system described here, the MyProxy server is the source of these credentials. When a user correctly authenticates the MyProxy server will generate a proxy for the user, based on either the user's real UK eScience certificate (if it has been previously uploaded) or a short-term, low-assurance certificate generated by its internal CA. The user interface can then use this proxy to authenticate the user to the head node of a resource and can also further delegate the certificate to the head node to allow other resources to be used.

2.1 Assurance

Assurance, in this context, is a rough measure of the extent to which authentication token can be trusted to represent the user's true, "real life" identity. The Grid community defines four assurance levels [10]:

- *Rudimentary*: there is essentially no *proof* of the user's identity;
- *Basic*: The applicant provides proof of identity to an agent of the issuing authority via a reasonably secure method, but may not have to appear in person;
- *Medium*: The applicant must appear in person to the agent, presenting adequate photographic proof of identity;
- *High*: Essentially as medium, except at the time of renewal, the applicant must present photo id again, and there are greater requirements regarding the protection of the user's private key.

To be internationally approved, the eScience CA has to operate to the medium assurance level. The CA and its verifying agents form a well-defined entity that can account for the flow and subsequent use of personal data throughout, from the first application to the final certificate expiry. In the terminology of the Data Protection Act [11], the CA (more specifically, the CA manager) is the *data controller*: the person who defines which data is necessary and required, what it is used for, and how it should be stored and managed. Furthermore, in the case of a problem such as misuse of a resource, the resource admin can report the user's DN to the CA and the CA can prove, using its personal data, who the user is in "real life."

With a credential conversion service, managing the user's data is *externalised*: it is in the hands of some other authority, external to the issuing authority and its agent — not just at the time of identity verification, but throughout the "lifetime" of the data. It

thus becomes harder for the service to even define, and much less guarantee, its assurance level. Some sites can generate authentication tokens from their payroll databases, but those are the minority. Most sites' databases have visitors and contractors, etc.

Worse yet, in the misuse case mentioned above, suppose a resource admin reports the DN to the credential conversion service admin, but the latter cannot provably tie the site authentication token to the user's real life identity. Only people with appropriate access to the site database (usually HR staff) can do that, and they probably will not, for data protection reasons, because the resource is external to their site. This aspect may make resource administrators less inclined to trust the credential conversion service.

Since we cannot say, even for our own site, what is the assurance defined by the site staff database, we, somewhat arbitrarily, refer to certificates created by the SSO service (from the MyProxy CA) as "low-assurance". The reader may find it helpful to compare it to basic assurance: in some sense, MyProxy is an agent which receives and validates an authentication token from the applicant, and issues a certificate.

Another potential problem with the SSO architecture is that each site has its own credential conversion service. This means that rather than reviewing and installing a single point of trust, the resource admin must now review policies of tens of services, and install them individually.

In theory, if all sites' services could agree to have:

- a common policy of adequate level (the policy will essentially be the "lowest common denominator," i.e., loose enough to accommodate all sites),
- a common root certificate with each site's service CA issued by that root,

then, because these services do not issue CRLs, it should be possible just to trust the common root. However, on the Grid, Globus still requires the signing policy files, to perform extra validation of the namespaces.

Thus, to install a credential conversion CA for each site, may be possible in a single (small-ish) country, or in projects with few collaborating sites, but on a larger, or international, scale it becomes difficult.

For these reasons, and also to provide host and service certificates, an SSO infrastructure cannot remove the need for a national CA. Even for a project the size of NGS, work is required to make future SSO infrastructures compatible; some of this work will be done in the context of the Shibboleth roll-out, but Shibboleth is not yet integrated with the Grid middleware.

3 Implementation

For Grid SSO in CCLRC we have leveraged the Kerberos/Active Directory [3] security system already present to allow users to log in using either a Kerberos token or their site password. This addresses two of the SSO use cases (one and three) mentioned in Section 1.2. The different schemes are employed based on whether or not the user has a valid Kerberos ticket present.

The SSO solution also must support users that have an X.509 certificate locally (e.g., as issued by the e-Science CA). This is the second use case mentioned in Section 1.2, and is addressed by allowing users to access certificates stored in a variety of formats.

The e-Science CA is a web-based CA: when users download their personal certificate, it is stored in the browser, and usually exported from the browser in PKCS#12 format. For Grid work, users would typically have to convert this format into PEM format at the command line, using relatively complex OpenSSL commands. However, this conversion is not required in the SSO project: it is possible to use the certificate in both formats, both PKCS#12 and PEM. It is even possible to use the certificate directly from the user's browser, thus making the exporting stage unnecessary. This once again follows the SSO philosophy, allowing users to authenticate to Grid resources with a minimum of effort.

The main components involved in the CCLRC SSO system are the MyProxy server and the Java GSI-SSHTerm, which forms the user interface component from Figure 1.

3.1 MyProxy Server

The standard MyProxy server distribution comes both with Kerberos (using SASL authentication) and CA support as compile time options. Two MyProxy server instances on the same machine are employed for our SSO solution, sharing the same certificate store. Both also have the build-in CA support enabled, again issuing certificates from the same root certificate. The first is a non-Kerberos MyProxy server (using the normal MyProxy port of 7512); this is linked through a PAM module to the site authentication system for site password access to the SSO facilities. A normal MyProxy server also allows users to upload real e-Science certificate proxies to the server using the standard tools. The second MyProxy server runs on port 7513 and uses the Kerberos/SASL authentication.

When a low-assurance certificate is required the MyProxy servers call out to a small program which is used to map a site user id to a DN. To do this it uses LDAP access to the site Active Directory and constructs the DN as follows: **/C=UK /O=eScienceSSO /OU=CCLRC /UID=***uid* **/CN=***firstName surname*. Resource administrators must explicitly trust the MyProxy CA by importing its root certificate to the resource and adding the SSO DNs of legitimate users to the resource's grid-mapfile.

3.2 Java GSI-SSHTerm

The Java GSI-SSHTerm[1] is used as the user interface in our SSO solution, providing GSI-SSH access to grid resources, through a platform independent Java SSH terminal. The original Java SSH terminal is an open source project hosted on Sourceforge. Jean-Claude Côte at NRC-CNRC then developed a GSI module for the SSH terminal, which we have extended and improved. The GSI module and our additions rely on the Globus Java Commodity Grid (CoG) libraries [12]. For the SSO project the MyProxy parts of the CoG libraries were modified to support the Kerberos based MyProxy servers through the addition of support for the SASL authentication method. The Java GSI-SSHTerm uses Kerberos authentication if the user has a Kerberos ticket and otherwise allows users to supply their password to access the MyProxy servers. However, making use of the Kerberos token requires Java 1.6, and only Java 1.4 and 1.5 are widely available. We have successfully used a beta release of Java 1.6.

Although this new functionality is exposed through the Java GSI-SSHTerm it could be used with any other Java based grid user interfaces and tools such as a proxy upload tool.

In addition, as previously mentioned in Section 3, the Java GSI-SSHTerm allows direct access to real e-Science certificates in a number of ways.

3.3 Renewal

When the terminal's proxy expires, the user is prompted to re-authenticate, either via MyProxy or via the user's local credentials. For longer running sessions where the user stays logged in but is absent from the terminal, command output is *not* lost. The terminal prompts the user for authentication, but continues to display the command output, but the user cannot type anything into the terminal until the authentication is successful. Of course, if the user authenticates with a local Kerberos ticket, it is sufficient that the terminal can pick that up. On Windows, for example, Windows renews the ticket automatically when necessary, and the terminal re-authenticates the user without any user intervention.

The proxy on the remote host, the Grid head node into which the user logs in via the terminal, can also expire. This proxy is created initially by the terminal, but is not renewed by the terminal in the current implementation. The user will have to re-authenticate to the MyProxy server to get a new proxy out of it.

3.4 File transfer

Grid access is more useful if the user can transfer data from and to the Grid head node within the session. The Java GSI-SSHTerm includes a graphical transfer interface that the user can use to upload and download files using the SFTP protocol (over GSI).

3.5 X forwarding and VNC client

The Java GSI-SSHTerm is capable of secure "X forwarding" to an X Window System server running on the same computer as the Java GSI-SSHTerm program. In other words, programs running on the head node can display their windows on the user's desktop machine.

Alternatively, users can use a Java VNC client built into the program to connect to a VNC server session running on the head node.

3.6 CCLRC Changes to the GSI-SSHTerm

To bring the SSHTerm and the GSI module into a form useful for Grid users we have had to both add a number of features and also perform hardening and bug-fixing throughout the code.

Within the main body of the SSHTerm code, changes included new build scripts, large changes to the X forwarding code to allow correct authentication under UNIX and to allow UNIX domain sockets to be used, improvement to protocol conformance in the choice of key exchange, cypher, compression, etc., algorithms (taking the responsibility for this bewildering decision from the users) and a redesign of the open connection dialog to allow easy connection for users connecting to Grid resources using default settings similar to default SSH settings.

In the area of GSI support the original authentication method implemented by Jean-Claude Côte could not authenticate users who did not know their username. To enable logon without a specified username we had to implement the "external-keyx" authentication algorithm. In addition, this authentication algorithm requires the "gss-group1-sha1-*" key exchange algorithm, which we also implemented. Other work in the area of GSI support included porting Jean-Claude Côte's code to a later version of

[1] http://www.grid-support.ac.uk/content/view/81/62/

the CoG kit and implementing methods to obtain proxies from additional sources (MyProxy servers, PKCS#12 files and browsers) and integrating these methods into a consistent and unified user interface.

These changes to the GSI-SSHTerm are in addition to the changes to the CoG libraries and the GSI-SSHTerm to allow Kerberos authentication to MyProxy servers.

3.7 Deployment

As we write this, our SSO solution is in the process of being rolled out within CCLRC and pilot users have been accessing the CCLRC SCARF cluster through the SSO solution using auto-generated low-assurance certificates. Further, through the Diamond-CCLRC SSO committee there are plans for our SSO solution to be used across the Diamond and ISIS facilities. Users can already use the SSO solution as a fast and convenient way of accessing any Grid resource (including the NGS) for which they are authorized, once they have uploaded a real UK e-Science proxy to the SSO MyProxy server.

4 Security Evaluation

From the point of view of the Resource Manager, our SSO solution is an implementation of the Grid Security Infrastructure specification as described by Von Welch *et al* [13]. Trust is established upon presentation by the user of an X.509 identity proxy certificate [14, 2]. As with choosing to trust a classic CA, the Resource Manager must choose to trust the SSO CA or another classic CA to enable SSO access for a user by installing the root certificate of the CA. Once this is done, the user is granted access to the resource once their DN is registered in the normal way.

4.1 SSO MyProxy Server

Our SSO solution requires short-lived X.509 proxy certificates (of real UK eScience certificates) to be stored on a MyProxy server unencrypted so authentication is performed transparently. The MyProxy server thus represents a potential security problem if it is not adequately secured. However, it must be noted that the lifetime of the proxy certificates stored on this server may be set to a short period of time. Ideally this should be no longer than the lifetime of the original authentication credential used to generate the proxy certificate, but this may not be practical or useful. In the case of users with certificates from a classic CA, this is the lifetime of their long-lived X.509 certificate. In the case of users without such a certificate, this lifetime is the lifetime of the user's Kerberos token (generated in the case of CCLRC by Microsoft AD Key Distribution Centre). It should be noted that this is 10 hours by default; as such, using our SSO solution without long-lived X.509 certificates may not be beneficial to users who need to launch Grid jobs that are likely to last longer than 10 hours.

4.2 Communication Channels

All communication between the user, the MyProxy server or MyProxy CA, the site authentication structure and Grid resource provider is performed over secured channels. This is done by making use of either Kerberos tokens belonging to the user or the MyProxy server, or X.509 certificates in the case of communication between the resource provider and the MyProxy server.

Using our SSO solution alleviates the requirement for users to authenticate themselves by entering a password if the user already possesses a Kerberos token, which could be transparently generated during login to the user's computer. This happens automatically with the Windows operating system if the computer is registered on the domain. In the case of Linux, this can be made possible using a solution such as Vintella. If this is not the case, a user can still gain access to the Grid resource provider, albeit not transparently, using our solution by providing their AD login details to the MyProxy CA server, which then retrieves a Kerberos token by calling out to the AD service solely for the purpose of generating a short term proxy certificate.

4.3 Java GSI-SSHTerm

The Java GSI-SSHTerm user interface is distributed both as a trusted applet and as a standalone application. In both cases the program needs read and possibly write access to the local disk. If the user has their certificate stored locally on disk or in a browser, the program needs to read files and libraries locally, which could include the execution of native code. If run for the first time on a computer, then the UK e-Science root certificate and signing policy is installed in the default location onto that computer, if these files have not already been installed. This is to facilitate running other Grid or CoG kit software in the future.

When run as an applet, such disk access can only be performed if the applet has been given explicit permission to do so by the user. This is done by accepting an object signing certificate issued by the UK e-Science CA (the ca-operator certificate). This gives the user the assurance that the program has not been tampered with, e.g. no Trojan has been intro-

duced to gain unauthorized access to the user's certificate. We plan to investigate the use of Java Web-Start technology in the future to give the similar level of assurance to users when running the program as a standalone application.

4.4 Certificate Security

The existence of inadequately protected X.509 certificates have in the past been a cause of security concern. Ideally the user should have a minimal number of certificates necessary to work with the Grid, and have a copy backed up in a secure offline location. Our SSO solution alleviates the need to have long term certificates issued by a classic CA to make use of local Grid facilities. However, if a user needs a long term certificate then the GSI-SSHTerm provides the additional security option of having the user's only certificate in their browser.

Unlike generating a proxy certificate in the traditional way using the MyProxy client tool, proxy certificates generated by the GSI-SSHTerm or retrieved by a MyProxy server are not stored as a file but rather in resident memory. This minimizes the possibility of a proxy certificate being intercepted on the client machine.

We assert that if reasonable effort is spent safeguarding the deployment of this solution and educating users to use it in a responsible manner, our SSO solution using Java GSI-SSHTerm is no less secure than existing Grid access mechanisms.

4.5 Password Security

A SSO infrastructure potentially *improves* the security of the authentication infrastructure:

- If users may remember fewer passwords – ideally, a single one – they are less likely to write them on notes and stick them to their monitor.
- If users are forced to remember more than one password, because systems are not integrated, they are very likely to reuse their passwords. This is particularly bad in the case of the federal (Active Directory or Kerberos) password, because site policy dictates that this password must be stored only in the central service (KDC). If users reuse *this* password, storing it in potentially less secure places, then site security may be compromised.
- The password is validated by a central service, which means that it can be checked for strength, according to site policy. This is not true for the password that protects the private key, or the browser's keystore, but at least those do not, hopefully, leave the user's desktop machine.

The security improvement is likely to be greater for security-novice or non-technical users who are not used to remembering complicated passwords.

5 Future Work

The work in SSO is progressing in a number of directions. As mentioned in Section 3.2 the CoG libraries, which have been modifed to support SASL/Kerberos, can be used in many other contexts. It is key that the vast majority of site Grid resources support single sign-on, in a consistant way, if the wider single sign-on project at CCLRC is to be successful, therefore work is in progress to use this technology in other Grid related user interfaces, for example with the Grid portals work at Daresbury Laboratory.

An additional area of future work is integrating Shibboleth into our SSO framework. JISC will be deploying a Shibboleth infrastructure [15] for UK academia. Meanwhile, we can build on work done in the various Shibboleth projects concurrent with our work, such as the ShibGrid project at Oxford/CCLRC and the Shebangs project at Manchester (see [16] for details of these and other Shibboleth/Grid projects). The ShibGrid project aims to provide Shibboleth access to NGS, and its architecture is similar to that of Figure 1, with Shibboleth cookies instead of Kerberos tokens. Thus, at least in theory, it should be easy to integrate Shibboleth authentication into the terminal. One potential complication is that the current versions of Shibboleth are entirely web-based: they depend on the HTTP protocol and redirects, etc.

6 Acknowledgments

The authors wish to thank the GOSC (Grid Operations Support Centre) for funding this work. They also wish to express their gratitude to all early users within CCLRC, and in particular those of the CCLRC SCARF cluster, who, led by Dr Duncan Tooke, have already put the terminal into production.

This document is typeset with LaTeX2e.

7 Conclusion

The usual definition of Single Sign-on (SSO) is often too narrow to be useful. In this paper, we start with what we see is a more useful definition of SSO based on requirements gathered from users who already use our Grid resources. Working to those definitions, we provide an implementation of a portable terminal to access Grid resources. We have taken an open

source project with GSI extensions, and integrated it with the site infrastructure using MyProxy. Our own software developments and additions to the terminal are of course also released under an open source licence. We have increased the ease-of-installation by deploying it both as a tarball, and as an applet.

We have described the architecture and how it ties the Grid and the site authentication infrastructure together. We have also described how it fits into the larger picture, with a national CA and a Shibboleth infrastructure, and why we see a need for all three (a CA, the Shibboleth infrastructure, and site credential conversion) in the future.

References

[1] David Byard and Jens Jensen. Single sign-on to the grid. In *Proceedings of the 2005 UK e-Science All Hands Meeting*, September 2005.

[2] S. Tuecke, D. Engert, I. Foster, M. Thompson, L. Pearlman, and C. Kesselman. Internet X.509 public key infrastructure proxy certificate profile. Request for Comments (RFC) 3820, June 2004.

[3] Robbie Allen, Joe Richards, and Alistair G. Lowe-Norris. *Active Directory*. O'Reilly, Sebastopol, California, USA, 3rd edition, January 2006.

[4] Jennifer G. Steiner, Clifford Neuman, and Jeffrey I. Schiller. Kerberos: An authentication service for open network systems. In *Proceedings of the USENIX Winter Conference, Dallas, Texas, USA*, pages 191–202, February 1988.

[5] Jim Basney, Marty Humphrey, and Von Welch. The MyProxy online credential repository. *Software: Practice and Experience*, 35(9):801–816, July 2005.

[6] EU Grid Policy Management Authority. Classic authentication policy profile. Available at *http://www.eugridpma.org/igtf/*, Sep 2005. (version 4.03).

[7] Robert Petkus. One-time-password integration at BNL. Available as *http://hepix.caspur.it/spring2006/TALKS/4apr.petkus.otpbnl.pdf*, April 2006. (Talk given at HEPiX conference).

[8] Jason Novotny, Steven Tuecke, and Von Welch. An online credential repository for the Grid: MyProxy. In *10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10), San Francisco, California, USA*, pages 104–114, August 2001.

[9] S. Santesson and R. Housley. Internet X.509 public key infrastructure authority information access certificate revocation list (CRL) extension. Request for Comments (RFC) 4325, December 2005.

[10] Randy Butler and Tony Genovese. Global grid forum certificate policy model. Available at *http://www.ggf.org/documents/GFD.16.pdf*, Jun 2003.

[11] Data Protection Act 1998. Available as *http://www.legislation.hmso.gov.uk/acts/acts1998/19980029.htm*, March 2000.

[12] Gregor von Laszewski, Jarek Gawor, Peter Lane, Nell Rehn, Mike Russell, and Keith Jackson. Features of the Java commodity Grid kit. *Concurrency and Computation: Practice and Experience*, 14:1045–1055, 2002.

[13] Von Welch, Frank Siebenlist, Ian T. Foster, John Bresnahan, Karl Czajkowski, Jarek Gawor, Carl Kesselman, Sam Meder, Laura Pearlman, and Steven Tuecke. Security for Grid services. In *12th International Symposium on High-Performance Distributed Computing (HPDC-12), Seattle, WA, USA*, pages 48–57, 2003.

[14] CCITT recommendation X.509: The directory - authentication framework. CCITT Blue Book, volume VIII, pages 48–81, 1988.

[15] Tom Scavo and Scott Cantor. Shibboleth architecture technical overview. Internet 2 document: draft-mace-shibboleth-tech-overview-02, June 2005. Available at *http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf*.

[16] Von Welch. Report for the GGF 16 BoF for grid developers and deployers leveraging shibboleth. Summary of BoF sessions at GGF 16, February 2006. Available at *http://grid.ncsa.uiuc.edu/papers/GGF16-Shib-BOF-Report.pdf*.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

A case for Shibboleth and grid security: are we paranoid about identity?

A paper for the UK e-Science All Hands Meeting, September 2006

Mark Norman,
University of Oxford

1. Abstract

The findings in this paper represent some of the output of the ESP-GRID project following the consultation of current grid users regarding the future nature of grid computing. The project found that there was a clear purpose for Shibboleth in a future grid and that, for the majority of users, this would be secure and improve their experience of grid computing. Client-based PKI remains suitable and desirable for Power Users and we must be careful of the means by which we mix these two access management technologies. PKI is currently used to define grid identities but these are problematically conflated with authorisation. The grid community should work harder to separate identity/authentication and authorisation. This paper also questions whether we need identity to be asserted throughout grid transactions in every use case. Currently, this is a solution to a security requirement: it should not be a requirement in itself. We propose that the grid community should examine methods for suspension of a rogue user's activities, even without identity being explicitly stated to all parties. The project introduced the concept of a Customer-Service Provider model of grid use and has produced demonstrators at the University of Glasgow.

2. Introduction

2.1. The ESP-GRID project

This paper represents some of the output of the Evaluation of Shibboleth and PKI for Grids (ESP-GRID) project (URL in References). The project also has thoughts and findings on the types of users who may populate a future grid and on the idea of a Customer-Service Provider model of grid use. These are found in a separate All Hands paper (Norman, 2006).

The ESP-GRID project has evaluated the access management requirements of grids both from the existing literature and the projected future set of users. It has also investigated the technologies available for policy management and looked at the concept of virtual organisations. Much of the technical output of the project has been in the form of demonstrators developed at the National e-Science Centre Hub at the University of Glasgow,UK (URL in References).

2.2. Grid security: what are we trying to secure?

Grid computing tends to be thought of as displaying a quite different threat model to that of other network environments (e.g. the world wide web). With grid computing the concept is that the user has some degree of control of the remote grid machine that she is accessing: instead of – for example – merely returning a document, she is able to take up the processor of the machine for an extended amount of time and she is usually able to modify the environment on that machine as well. A rogue user in such a situation clearly could pose a far greater threat than in more traditional 'Internet' situations. Alternatively, we have argued (Norman, 2006) that in the near future – if grid computing is truly successful – most users may access grid services in a very controlled manner that has many similarities with the world wide web. Such users are not likely to be able to modify the environment on the grid machine and may be limited to very predictable actions.

Therefore, most activities on a grid may pose a much lower threat to grid machines than the activities that dominate today. For the sake of this paper, let us assume that we have a mixed economy of users: some exerting relatively deep control over distant grid machines and many users with little scope or interest in modifying the computing environment or how the jobs run on the grid.

2.3. Sections of this paper

Within this paper, we examine identity management and who is best to take on this task. This is followed by an examination of the perceived requirement for constant identity assertion throughout the grid and the 'case for Shibboleth'.

3. On the grid is it appropriate to devolve identity management?

Traditionally, user 'identities' have been managed in the higher education community on a per-institution (organisation) basis. There has been little drive to be very rigorous about checking real-world identity accurately when issuing identity credentials at such organisations for the first time, although it is likely that these procedures have been better than many believe. Those working in a stricter (usually PKI) culture may consider these procedures to be inferior. However, there are strengths and weaknesses to both the (usual) PKI approach and to the per-institution approaches.

3.1. In the UK, we are already trusting the old ID-establishment processes

At present, an applicant for a digital certificate only needs to present *some form* of photo ID (undefined in the UK e-Science Grid CA - Certificate Policy and Certification Practices Statement). Usually this is taken as a person's university card. This means that we are trusting the procedures for issuing the university card in the first place. It follows that the original choice for choosing client-based PKI for grid security is somewhat flawed, as the strongest part – the greatest benefit – of PKI: the establishment of a long-term, highly trustworthy, ID is compromised. This is an argument for another place, however.

A further difficulty with mixing old university procedures and newer, very centralised, PKI procedures can be summed up in this scenario:

> *Post-grad A. Newman begins work at Cotswolds University. He finds he needs a digital certificate for some of his grid-based research. He talks to his local registration personnel about this who know nothing of the "grid" and then finds he has to travel to his local Registration Authority at Oxford University, after applying on-line. He travels to Oxford and presents his 'Cotswolds Card'and the RA grants his certificate request.*[1] *A little later, it turns out that Newman is a thief and a fraudster and Cotswolds University revokes all of his university accounts, swipe cards etc. etc. Unfortunately, the good registration folks at Cotswolds don't have anything to do with e-Science (they haven't been on the RA training course) and therefore Newman is allowed to keep his digital certificate for the rest of the year.*

It is clearly better that the registration or personnel people closest to the user should look after the identity of that user. PKI is usually over-centralised and managed at a very remote, often national, level, as in the UK. This is highly problematic. This case

[1] Assume that the "Cotswolds Card" is his university ID. However, a lovely twist to this story would be that he (theoretically) could have used a "Cotswolds Card" that was issued by his local swimming pool (with inadequate ID checks), but that still contained his photograph.

has been made at greater length elsewhere (Norman, 2005).

3.2. Where PKI should work in managing identities

We should address the concepts of 'identity' and 'identity provision' and the management of identity. In a perfect on-line world, identity management would be completely separated from authorisation. However, at present, this is rarely the case. Grids using client digital certificates, for example, tend to have the *Organisation*, to which the person belongs, included on the certificate. The certificates are issued typically for a year and users are able to obtain a certificate only if they are a member of a particular research or grid community. All of these factors are attributes associated with authorisation decisions. If such authorisation decisions were handled quite separately from the identity token (e.g. digital certificate) then users would be able to keep the token for life. It would not need to be managed, except for the instances where it was issued mistakenly or wrongly or if it had been 'stolen' by another entity. The person will still be the same entity in ten years' time, even if she had undergone a sex change, been convicted of defrauding other grid users etc. etc. Her identity would not have changed, but her authorisation attributes certainly would!

3.3. Identity and attribute management

Currently, it is easier to combine identity, authentication and authorisation to some degree. Identity tokens (accounts, user names, digital certificates etc.) are issued by organisations such as education establishments and it is these same organisations that help the resource providers (e.g. grid nodes) to make authorisation decisions about users. This need not be the case, but if this 'identity problem' were to be solved then the problem would just transform to a problem of managing authorisation-associated attributes.

4. The grid requirement for identity

4.1. Emotional security

When discussing and planning security mechanisms it is always surprising how often one's emotions can cloud the issues. We tend to assume that a system is more secure if the users and other entities therein are always explicitly and fully identified (i.e. there are logs of identities associated with most actions). This is only true if those identities may be checked accurately, the data is current and the authorisation is similarly accurate. Without those caveats, explicit identities can give a thoroughly false sense of security.

Emotionally, we always want to know "who" the user is, in case they do something wrong. Actually, as the "who" is really quite difficult to check and the authorisation credentials even more difficult, it should be the "can I trace this user easily if he does something wrong" that should be far more important, as should the concept of, "actually, I don't mind who this is right now, just as long as I'm fairly sure that they are authorised". But those don't give us a warm feeling of security. They are, nevertheless, far more secure than relying on poorly maintained identity (mixed with authorisation) information.

Bruce Schneier writes about the great insecurity of relying too much on ID (Schneier 2004a) and also gives examples of where this can lead to surprisingly (and possibly unexpected) reduced levels of security (Schneier 2004b). These examples include airline traveller programs whereby travellers can register beforehand, go through an identity check and thereafter reduce the chance of having their baggage searched at airports: clearly a first-time terrorist gains an advantage by such a situation. Schneier cites excellent examples of terrible security which makes people feel

better and points out how good security may seem counter-intuitive until looked at in depth. With regard to the use of ID, he rightly suggests that if you make something easier (i.e. lower security) if ID is used, then the bad guys will just get ID. And the rest of us are left not paying enough attention to security because the ID has given us a false sense of security.

4.2. Do we need identity throughout, for every service?

Currently, grids' supposed requirement for 'up front' identity assertion throughout may be exaggerated. Some services certainly need to know the identities of users. However, many do not: the hard requirement for identity has probably come about as it is a *solution* to the requirement to suspend the activity of wrong-doers or for when certain users' credentials have been stolen.

4.3. Rapid suspension and slower identification/revocation?

Explicit identity may be useful at times, but it is clearly secondary in importance to a guaranteed method of quickly detecting wrong-doers and of removing their privileges. This may be achieved with or without knowing identity 'up front' or by logging permanent identities. Therefore, the main requirement should be the detection of misuse or security breaches and the quick tracing of the identity of the user, rather than constant logging of identity.

The real requirements are probably for:

- good authorisation procedures;
- quick detection of wrong-doers;
- rapid suspension of rights, possibly throughout the grid;
- (in most cases) the rapid revocation or suspension of ongoing jobs throughout the grid;
- an investigation into the activities of the individual.

These requirements are expanded upon and tied to the Customer-Service Provider model (see 5.1 below).

5. The case for Shibboleth

5.1. The C-SP model will dominate

As outlined elsewhere (Norman, 2006), it is very likely – on many mature production grids – that the majority of users will benefit from the power of grid computing through an application-interface on a server: for example, via a web portal. We have called this the *Customer-Service Provider model* (C-SP model). With such a restrictive point in terms of the possible range of actions that a user can undertake, the use of Shibboleth to enable authentication and authorisation is highly appropriate. The use of the grid via the C-SP model is summarised in Figure 1. The abbreviations SEU (Service End User), IdP (Identity Provider), and SP (Service Provider) are described at greater length in Norman (2006). The SP uses a set of host certificates to interact with the grid. The grid machines could cause the revocation of one or more of these host certificates if an attack were suspected and/or the SP and IdP could be made to suspend a user's activities automatically in such a case. This could then, typically, be followed by human attentions within the IdP and SP to identify the user and investigate which actions should be taken.

It is a widely-held principle that the organisations interacting most frequently with the end user are the most appropriate to manage their identities and/or their most common authorisation attributes. Conversely, exceptions to this exist in two main areas:

- If it were possible to truly separate authentication from authorisation on the grid, there is little reason

why long term identity tokens could not be issued. This would then mean that authentication could take place in a variety of places.

- Authorisation attributes may also be held with virtual organisations and a secondary query may be necessary.

Nevertheless, in the first exception cited above, it may still be most convenient for SEUs to be authenticated at their home organisation (IdP) for single sign-on reasons. Similarly, it may be convenient for the virtual organisation to allow authentication at the IdP or the SP before releasing the attribute information.

If we put the above two exceptions aside, then Shibboleth (http://shibboleth.internet2.edu/) is a good fit for devolving authentication and much of the management of authorisation attributes. Shibboleth would provide a useful single sign-on (like) experience for the user: he would only need to authenticate at his home organisation. This would benefit him in terms of having to learn only one sign-on interface, and would place the task of managing identities and attributes with the most appropriate organisation.

Shibboleth may not be appropriate if identities are established long-term, although the authentication of these identities may sit well with the home organisation, nevertheless. It is also likely that a forthcoming release of the Shibboleth software will be extended to accept authentication within one organisation and the retrieval of attributes from another (virtual) organisation.

5.2. Demonstration of the C-SP model with Shibboleth

The BRIDGES, DYVOSE, VOTES (URLs in References) and ESP-GRID projects have produced a Shibboleth-enabled portal with which to authenticate and authorise people to access a variety of applications. This activity proves that Shibboleth and the grid can interoperate, but it avoids the issues of supporting Power Users. These issues may be unimportant unless the number of Power Users grows greatly. The need for something very easy to use for good uptake by researchers was discovered early on in the BRIDGES project, in particular, by the developers at Glasgow, both in terms of access management (and the need to avoid client digital certificates) and in a clean, easy, "Google-like" interface (Sinnott, 2006).

5.3. Most users are not Power Users

Missing from Figure 1 are the other types of grid user (described and discussed in Norman, 2006). These include the most common type of user that exists today. Such users, who typically work at the command line, write and/or compile code and often wish to modify the environment at a remote grid node, we have termed 'Power Users' in the ESP-GRID project. Power Users are likely to be able to tolerate the difficulties of working with PKI and any security advantage derived from the use of PKI is of benefit to the grid as such users pose a greater threat to an individual grid node.

5.4. Can Power Users benefit from Shibboleth?

There are several initiatives under way that are attempting, in different (and similar) ways to bring together the security of PKI and the ease of use of Shibboleth (GridShib, SHEBANGS, ShibGrid, MAMS, SWITCHaai, among others).

Some approaches need a mapping between an individual's Distinguished Name (DN) on his digital certificate and an 'attribute' that the user's home enterprise directory (or Attribute Authority – AA – in Shibboleth terms) can manage and supply, when requested. This would allow the identity to be the same, however the authentication were performed (e.g. via username/password and Shibboleth or via presentation of a digital certificate). Other

approaches (e.g. GridShib) mandate the use of a certificate for authentication but then use a Shibboleth AA for authorisation purposes. Some of these projects are also examining using Shibboleth and institutional single sign-on mechanisms to release digital certificates for use on the grid, but see *6.3-Mixing trust models*, below. There is much effort being applied to the Shibboleth-enabling of MyProxy servers which may prove very fruitful.

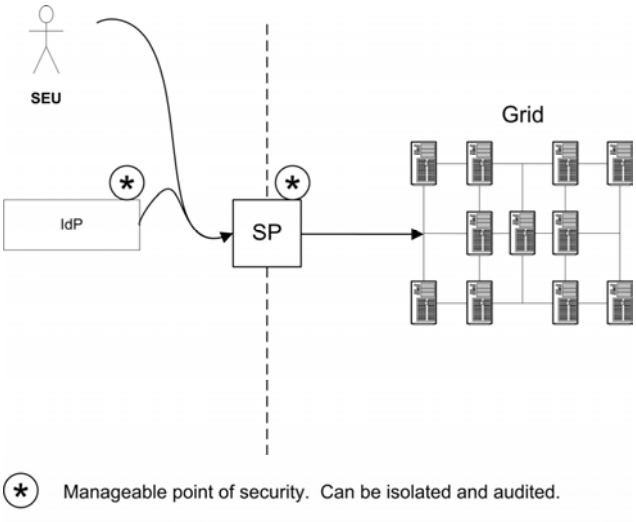


Figure 1 The C-SP model of access to the grid: the SEU is authenticated by the IdP (trusted by the SP) and the SP accesses the grid via a host certificate.

6. The cases against Shibboleth

6.1. Power Users

Power Users are probably an inappropriate group to benefit from the Shibboleth model of accessing the grid. If we accept that the use of PKI is beneficial to the grid as a whole, then it is this set of users who should be using client digital certificates, as today. There may be some isolated benefits to these users through using Shibboleth, such as 'away from home' access to pre-prepared proxy certificates or access to basic level assurance certificates for some tasks. If, however, our prediction proves to be incorrect and grid use does not grow via the C-SP model, and Power Users remain the dominant group, Shibboleth may be of limited benefit.

6.2. Delegation

An extension (RFC 3820) to the original standard (from RFC 2459 profiling X.509 version 3 certificates) allows for delegation via proxy certificates. This work has arisen largely due to the use of PKI on grids and Globus-based grids in particular. There are some 'philosophical' difficulties with such an approach; notably where the (proxy) private key does not remain in the sole control of the original user. However, this activity has proved a way forward for delegation on grids and is clearly a mechanism for *constraining* delegation. Shibboleth in itself does not provide a mechanism for delegation. (Shibboleth is based upon machine-to-machine trust and is, to some extent, incompatible with this concept, but see next section for an analysis of trust). Some would say that the use of proxy certificates gives rise to the situation of machine-to-machine trust and therefore, the need for this kind of delegation may be inappropriate, but this is an argument outside the scope of this paper.

6.3. Mixing trust models

The routes of trust for Shibboleth and for client-certificate PKI are a little different. In PKI, a user has a certificate and invokes it directly when interacting with a grid machine. If we ignore the checking of signatures and CAs, the trust is from human to machine (the human with the certificate does not need to rely upon another entity or machine for her certificate to be believed and trusted for authentication). This PKI represents 'human to machine trust' for the authentication/identification step.

Shibboleth requires the user to log in at his home organisation's identity provider (usually a web single sign-on interface). The assertion that "this user has been authenticated" therefore comes from a machine. Therefore Shibboleth represents 'machine-to-machine trust' for the authentication/identification step.

We need to take extra care how we combine these two methods. A mixing of the two trust models is problematic and – at the very least – brings down the overall security (or assurance) level to that of the least secure component. If we were to use Shibboleth to make the user experience with PKI less onerous, we are certainly reducing the assurance level of the assertion. (However, if the Shibboleth home organisation identification and authentication procedures were very robust for that user, it may not reduce the assurance level that much). This issue is in addition to the security challenge posed by the complexity of mixing two access management 'systems': the resulting system may be very complex and the more complex something becomes, the more likely it is to develop security problems.

It could be argued that, if we wish to use Shibboleth, then we should avoid the use of client-based PKI completely: the user could employ Shibboleth to mediate authentication and authorisation and then a 'gateway' machine could be trusted by the PKI-based grid. This is, effectively, the C-SP model.

7. Summary

Some of the outcomes of the ESP-GRID project include that PKI is used at the moment to manage identities, but that these identities are problematically conflated with authorisation. In the UK and elsewhere, our current implementation of client-based PKI is very good at establishing identities, but is very poor at managing authorisation. The grid community should work harder to separate these two.

We should also question whether we need identity to be asserted throughout grid transactions in every use case. Identity being asserted and logged 'up front' before every transaction gives us a *feeling* of security. The real need is for rapid suspension of the rogue user-initiated activity and the later revocation of credentials and/or rights: people have confused this requirement with the current *solution* of identity assertion throughout the grid.

Shibboleth is a great opportunity to allow the appropriate people to manage identities and authorisation-enabling attributes. It is certainly worth pursuing in the grid world: it could have the benefit of increasing the level of security on the grid as well as the ease of use for non-computer technical users.

The ESP-GRID project postulates that, in order for the grid to scale, some sort of Customer-Service Provider arrangement is necessary to enable the new users who are not expert computer scientists. This C-SP model lends itself to Shibboleth very well but, equally, the authentication point could be at the service provider portal instead.

The project has worked with the National e-Science Centre at Glasgow University to produce some demonstrators which are good examples of both the use of Shibboleth and grid and of the C-SP model.

Mixing Shibboleth and client-side PKI for grid users is difficult and potentially insecure, although there will be cases where

it is useful and appropriate. Indications from the ESP-GRID project are that client-based PKI is appropriate for grid Power Users (the current majority of grid users), but that Shibboleth, combined with their local institutions' single sign-on technologies would benefit the vast majority of the future End Users.

8. References

BRIDGES (Biomedical Research Informatics Delivered by Grid Enabled Services) project web site http://www.brc.dcs.gla.ac.uk/projects/bridges/

Certificate Policy and Certification Practices Statement for the UK e-Science Grid CA http://www.grid-support.ac.uk/ca/

DYVOSE (Dynamic Virtual Organisations in e-Science Education) project web site http://labserv.nesc.gla.ac.uk/projects/dyvose/

ESP-GRID (Evaluation of Shibboleth and PKI for Grids) project web site http://www.oesc.ox.ac.uk/activities/projects/eprojects/esp-grid/index.xml

GridShib project web site http://gridshib.globus.org/

MAMS (Meta Access Management System) project web site https://mams.melcoe.mq.edu.au/

Norman, M.D.P. (2005) The case for devolved authentication: over-centralised security doesn't work. JISC Core Middleware: developments within Security and Access Management, 20 October 2005. http://www.dcoce.ox.ac.uk/docs/JiscNeSCMiddwareBriefingOct05.pdf.

Norman, M.D.P. (2006) Types of grid users and the Customer-Service Provider relationship: a future picture of grid use. Proceedings of the 2006 UK e-Science All Hands Meeting

Schneier, B. (2004a) San Francisco Chronicle, February 3, 2004 http://www.schneier.com/essay-008.html.

Schneier, B. (2004b) Boston Globe August 24, 2004 http://www.schneier.com/essay-051.html.

SHEBANGS (Shibboleth Enabled Bridge to Access the National Grid Service) project web site http://www.sve.man.ac.uk/Research/AtoZ/SHEBANGS

Sinnott, R (2006) Development of Usable Grid Services for the Biomedical Community. Proceedings of *Designing for e-Science: Interrogating new scientific practice for usability, in the lab and beyond*' workshop at the UK National e-Science Centre, January 25-26, 2006.

SWITCHaai web site http://www.switch.ch/aai/

VOTES (Virtual Organisations for Trials and Epidemiological Studies) project web site http://labserv.nesc.gla.ac.uk/projects/votes/

9. Acknowledgements

The ESP-GRID project is funded from the Joint Information Systems Committee (JISC) and the authors are grateful for the support of the Core Middleware: Technology Development Programme.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Service-Oriented Matchmaking and Brokerage

Tom Goodale[1], Simone A. Ludwig[1], William Naylor[2], Julian Padget[2] and Omer F. Rana[1]
[1]School of Computer Science/Welsh eScience Centre, Cardiff University
[2]Department of Computer Science, University of Bath

Abstract

The GENSS project has developed a flexible generic brokerage framework based on the use of plug-in components that are themselves web services. The focus in GENSS has been on mathematical web services, but the broker itself is domain independent and it is the plug-ins that act as sources of domain-specific knowledge. A range of plug-ins has been developed that offer a variety of matching technologies including ontological reasoning, mathematical reasoning, reputation modelling, and textual analysis. The ranking mechanism too is a plug-in, thus we have a completely re-targettable matchmaking and brokerage shell plus a selection of useful packaged behaviours.

1 Introduction

How does the e-scientist or the e-scientist's software agent find the web service that does what they want? In practice, the reality for the e-scientist may be more the result of social interaction than scientific evaluation. While collegial recommendation, as an approach, has a number of positive attributes, it also underlines the weaknesses of current service description languages and service discovery mechanisms if a user prefers to use other means to find the "right" web service. To facilitate the re-use of generic components and their combination with domain specific components, a new low-overhead approach to building matchmakers and brokers is required—and one that gains leverage from the currency of the grid: web services while also bringing the process and the control closer to the user. Significant work has already been undertaken to support information services, such as the Globus MDS, LDAP and recently registry services such as UDDI. Most of these systems however are based on an "asymmetric" relationship between a client and a provider – generally requiring the client to make query to a service provider, and the provider enforcing some policy after a suitable service has been discovered. Each of these systems are also restricted by the types of queries that they can support.

In this paper we describe an architecture that simplifies the deployment of bespoke matchmakers and brokers. The matchmaker is comprised of re-usable components, the use of which is demonstrate through a set of examples. Whether a user wants the function of a matchmaker—to find suitable candidate services—or of a broker—to select from the candidate services and invoke or even construct a workflow—depends on just how much the user wishes to trust in the intelligence of matching and ranking mechanisms. This is somewhat similar to hitting the "I'm feeling lucky" button in Google, except here the user is committing to the use of a grid resource and that may have cost implications.

The flexibility of the brokerage framework stems from the fact that its architecture involves a component based approach, which allows the integration of capabilities through the use of web services. Thus, constructing a new broker becomes a matter of composing a workflow involving: (i) a range of sources of service descriptions; (ii) a range of matching services that will accept a service request and a service description and output some information about the relationship between the two; (iii) a ranking service that can order the service matching results information to determine the best fit; (iv) a service to invoke the selected service and deliver the results. For the user that would prefer greater control, instead of a ranking service, there could be a presentation service that contacts the user to display a list of options from which they may select. Whichever option is taken, the broker components employed, and other decision-making data may be recorded as provenance data in the answer document. It is also crucial to be able to provide feedback about why a particular service did or did not match—a form of explanation—since it is not just a matter of the presence or absence of a keyword as it is for Google finding and ranking a page. In the same way that humans choose carefully and subsequently refine their inputs to Google, we may expect users to want to do the same in identifying web services.

But how can a user express what they want of a web service? Keywords might help narrow down the search but they do not offer a language for describing the compatibility requirements, such as what inputs and what outputs are wanted. Furthermore a statement of the signature of a service says next to nothing about the actual function; for that we require the statement of relationships between the inputs and outputs, or more generally, statements of pre- and post-conditions. The reasonable conclusion is that we need a combination of syntactic, semantic and even social mechanisms to help identify and choose the right services.

We can therefore observe that each service will have a functional interface (describing the input/outputs needed to interact with it and their types) and a non-functional interface (which identifies annotations related to the service made by other users and performance data associated with the service). Being able to support selection on both of these two interfaces provides a useful basis to distinguish between services.

Even when a service (or a composition of a set of services) has been selected, it is quite likely that their interfaces are not entirely compatible. Hence, one service may have more parameters than another, making it difficult to undertake an exact comparison based just on their interfaces. Similarly, data types used within the interface of one service may not fully match those of another. In such instances, it would be necessary to identify mapping between data types to determine a "degree" of match between the services. Although the selection or even the on-the-fly construction of shim services is something that could be addressed from the matchmaking perspective [6, 10], we do not discuss this issue further in this paper.

The remainder of the paper is laid as follows: (i) in the next-but-one section (3) we describe the architecture in detail and the design decisions that lead to it (ii) this is followed by a description of the range of plug-ins that have been developed during the MONET and GENSS projects and that are now being integrated through the KNOOGLE project (iii) the paper concludes with a survey of related work and a brief outline of developments foreseen over the next year.

2 eScience Relevance

The GENSS project has developed a flexible generic brokerage framework based on the use of plug-in components that are themselves web services. The focus in GENSS has been on mathematical web services, but the broker itself is domain independent and it is the plug-ins that act as sources of domain-specific knowledge. Thus, bespoke matchers/brokers can be created at will by clients, as we demonstrate later in conjunction with the Triana workflow system. Each broker takes into consideration the particular data model available within a domain, and undertakes matching between service requests and advertisements in relation to the data model. The complexity of such a data model can also vary, thereby constraining the types of matches that can be supported within a particular application. Within GENSS and its predecessor project MONET, a range of plug-ins were developed that offer a variety of matching technologies:

1. **Ontological reasoning:** input/output constraints are translated to description logic and a DL reasoner is used to identify query to service matches,
2. **Mathematical reasoning:** analyses the functional relationship expressed in the pre-conditions and effects of the service (can also be applied to service composition),
3. **Reputation modelling:** recommendations from the user's social network are aggregated to rank service recommendations,
4. **Textual analysis:** natural language descriptions of requests and services are analysed for related semantic information.

The purpose of the framework is to make brokerage technology readily deployable by the non-specialist, and more importantly, an effective but unobtrusive component for e-Science users. Specifically this indicates the following modes of use:

1. As a pre-deployed broker in a work-flow demonstrating pre-packaged functionality and utility
2. As a bespoke broker using pre-defined plug-ins demonstrating the construction of a new broker from existing services
3. The authoring and/or packaging of plug-in services other than those described above, demonstrating the flexibility and interoperability of the architecture
4. The exploration of meta-brokerage, where the web service description of broker components are published and selected by a meta-broker to instantiate new special-purpose brokers.

3 Service-oriented Matchmaking

3.1 Matchmaking Requirements

We begin by reiterating the basic requirements for matchmaking:

1. Sufficient input information about the task is needed to satisfy the capability, while the outputs of the matched service should contain at least as much information as the task is seeking, and

2. The task pre-conditions should at least satisfy the capability pre-conditions, while the post-conditions of the capability should at least satisfy the post-conditions of the task.

These constraints reflect work in component-based software engineering and are, in fact, derived from [23]. They are also more restrictive than is necessary for our setting, by which we mean that some inputs required by a capability can readily be inferred from the task, such as the lower limit on a numerical integration where by convention this is zero, or the dependent variable in a symbolic integration of a uni-variate function. Conversely, a numerical integration routine might only work from 0 to the upper limit, while the lower limit of the problem is non-zero. A capability that matches the task can be synthesised from the composition of two invocations of the capability with the fixed lower limit of 0. Clearly the nature of the second solution is quite different from the

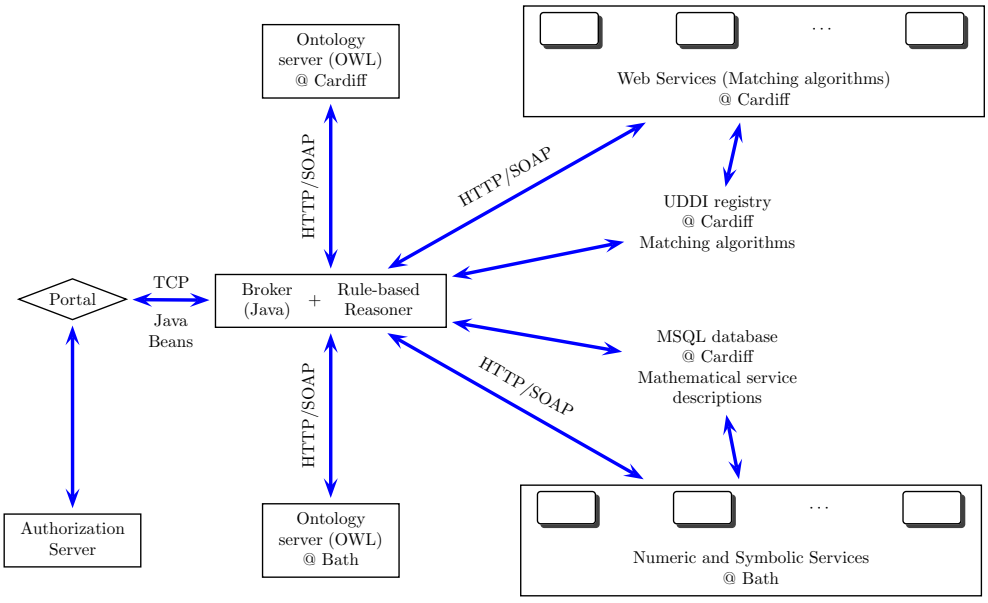


Figure 1: Architecture

first, but both serve to illustrate the complexity of this domain. It is precisely this richness too that dictates the nature of the matchmaking architecture, because as these two simple examples show, very different reasoning capabilities are required to resolve the first and the second. Furthermore, we believe that given the nature of the problem, it is only very rarely that a task description will match exactly a capability description and so a range of reasoning mechanisms must be applied to identify candidate matches. This results in:

> **Requirement 1:** A plug-in architecture supporting the incorporation of an arbitrary number of matchers.

The second problem is a consequence of the above: there will potentially be several candidate matches and some means of indicating their suitability is desirable, rather than picking the first or choosing randomly. Thus:

> **Requirement 2:** A ranking mechanism is required that takes into account pure technical (as discussed above in terms of signatures and pre- and post-condition) and quantitative and qualitative aspects—and even user preferences.

3.2 Matchmaking Architecture

Our matchmaking architecture is shown in Figure 1 and comprises the following:

1. The Authorization service, Portal: these constitute the client interface and are employed by users to specify their service request.
2. The Broker and Reasoner: these constitute the core of the architecture, communicating with the client *via* TCP and with the other components *via* SOAP.
3. The matchmaker: this is in part made up of a reasoning engine and in part by the matching algorithms, which define the logic of the matching process.
4. Mathematical ontologies: databases of OWL based ontologies, derived from OpenMath Content Dictionaries (CDs), GAMS (Guide to Available Mathematical Software) *etc.* developed during the MONET [15] project.
5. A Registry Service: which enables the storage of mathematical service descriptions, together with the corresponding endpoint for the service invocation.
6. Mathematical Web Services: available on third party sites, accessible over the Web.

There are essentially two use cases:

Use Case 1: *Matchmaking with client selection:* which proceeds as follows:

1. The user contacts the matchmaker.
2. The matchmaker loads the matching algorithms specified by the user via a look-up in the UDDI registry. In the case of an ontological match a further step is necessary. This is, the matchmaker contacts the reasoner which in turn loads the corresponding ontology.
3. Having additional match values results in the registry being queried, to see whether it contains services which match the request.
4. Service details are returned to the user via the matchmaker.

The parameters stored in the registry (a database) are service name, URL, taxonomy, input and output signatures,

pre- and post-conditions. Using contact details of the service from the registry, the user can then call the Web Service and interact with it.

Use Case 2: *Brokerage:* where the client delegates service selection via a policy statement. This proceeds essentially as above except that the candidate set of services is then analysed according to the client-specified policy and one service is selected and invoked.

Details of the various components of the architecture are discussed in [13].

4 Workflow integration

Workflow-based tools are being actively used in the e-Science community, generally as a means to combine components that are co-located with the workflow tool. Recently, extensions to these tools which provide the ability to combine services which are geographically distributed have also been provided. To demonstrate the use of the matchmaker as a service, we have integrated our Broker with the Triana workflow engine.

4.1 Triana

Triana was initially developed by scientists in GEO 600 [7] to help in the flexible analysis of data sets, and therefore contains many of the core data analysis tools needed for one-dimensional data analysis, along with many other toolboxes that contain units for areas such as image processing and text processing. All in all, there are around 500 units within Triana covering a broad range of applications. Further, Triana is able to choreograph distributed resources, such as web services, to extend its range of functionality. Additional web service-based algorithms have also been added recently to Triana to support data mining [17]. Triana may be used by applications and end-users alike in a number of different ways [21]. For example, it can be used as a: graphical workflow composition system for Grid applications; a data analysis environment for image, signal or text processing; as an application designer tool, creating stand-alone applications from a composition of components/units; and through the use of its pluggable workflow representation architecture, allowing third party tool and workflow representation such as WSDL and BPEL4WS.

The Triana user interface consists of a collection of toolboxes containing the current set of Triana components and a work surface where users graphically choreograph the required behaviour. The modules are late bound to the services that they represent to create a highly dynamic programming environment. Triana has many of the key programming constructs such as looping (do, while, repeat until etc.) and logic (if, then etc.) units that can be used to graphically control the dataflow, just as a programmer would control the flow within a conventional program using specific instructions. Programming units (i.e. tools) include information about which data-type objects they can receive and which ones they output, and Triana uses this information to perform design-time type checking on requested connections to ensure data compatibility between components; this serves the same purpose as the compilation of a program for compatibility of function calls.

Triana has a modularized architecture that consists of a cooperating collection of interacting components. Briefly, the thin-client Triana GUI connects to a Triana engine (Triana Controlling Service, TCS) either locally or via the network. Under a typical usage scenario, clients may log into a TCS, remotely compose and run a Triana application and then visualize the result locally – even though the visualization unit itself is run remotely. Alternatively, clients may log off during an application run and periodically log back on to check the status of the application. In this mode, the Triana TCS and GUI act as a portal for running an application, whether distributed or in single execution mode.

4.2 Triana Brokering

To support matchmaking for numerical services in Triana, the broker has been included within Triana in two ways:

1. A service in the toolbox: as illustrated in figure 2, the broker is included as a “search” service within the Triana toolbox. In order to make use of the service, it is necessary for a user to instantiate the service within a workflow, including a “WSTypeGen” component before the service, and a “WSTypeViewer” after the service. These WSType components allow conversion of data types into a form that can be used by a web service. Double clicking on the WSTypeGen component generates the user interface also shown in figure 2, requiring a user to chose a match mode (a “structural” mode is selected in the figure), specify pre- and post-conditions, and the query using OpenMath syntax. Once this form has been completed, hitting the “OK” button causes the search service to be invoked, returning a URL to the location of a service that implements the match. If multiple matches are found, the user receives a list of services and must manually select between them (as shown in figure 3). If only a single service is found, the location of a WSDL file may be returned, which can then be used by a subsequent service in the workflow (if the matchmaker is being used as a broker).
2. A menu item: in this case, the search service is invoked from a menu item, requiring the user to complete a form, as shown in figure 3. The user specifies the query using the form also shown in the figure, and selects a matching mode (a “Structural match” is selected in the figure). The result is displayed in a separate window for the user to evaluate. In this instance, it is not necessary to match any data types before and after a match service, and some additional components will need to

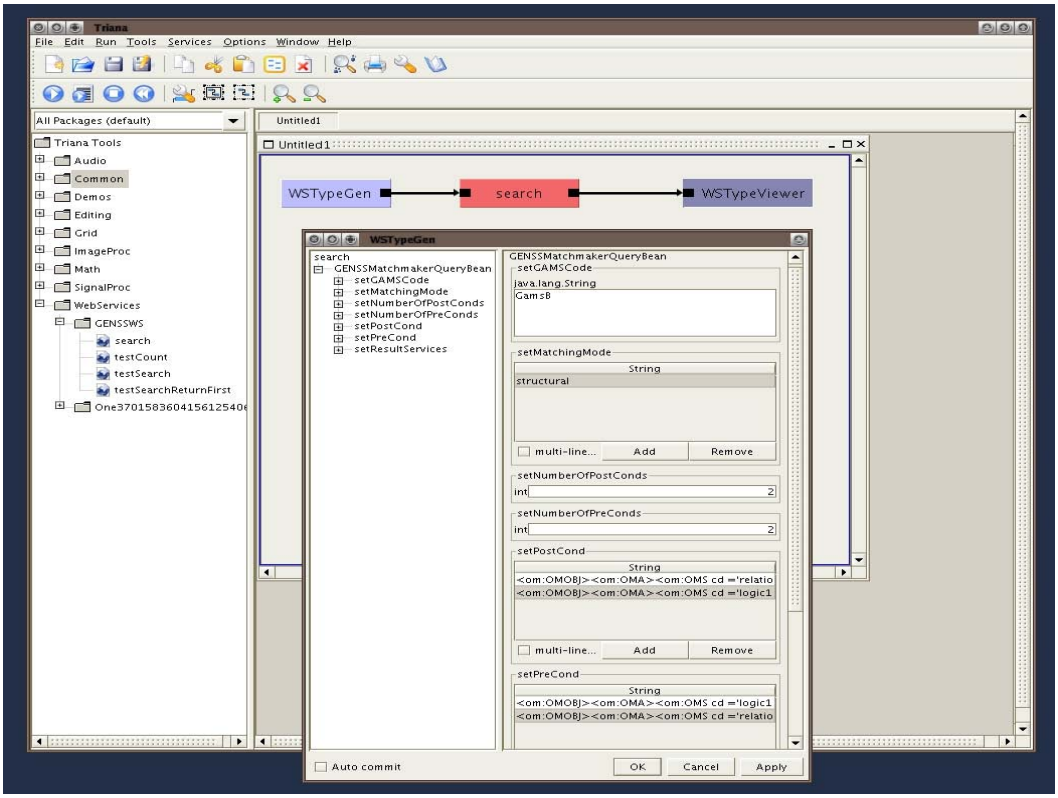


Figure 2: A Triana workflow with matchmaker

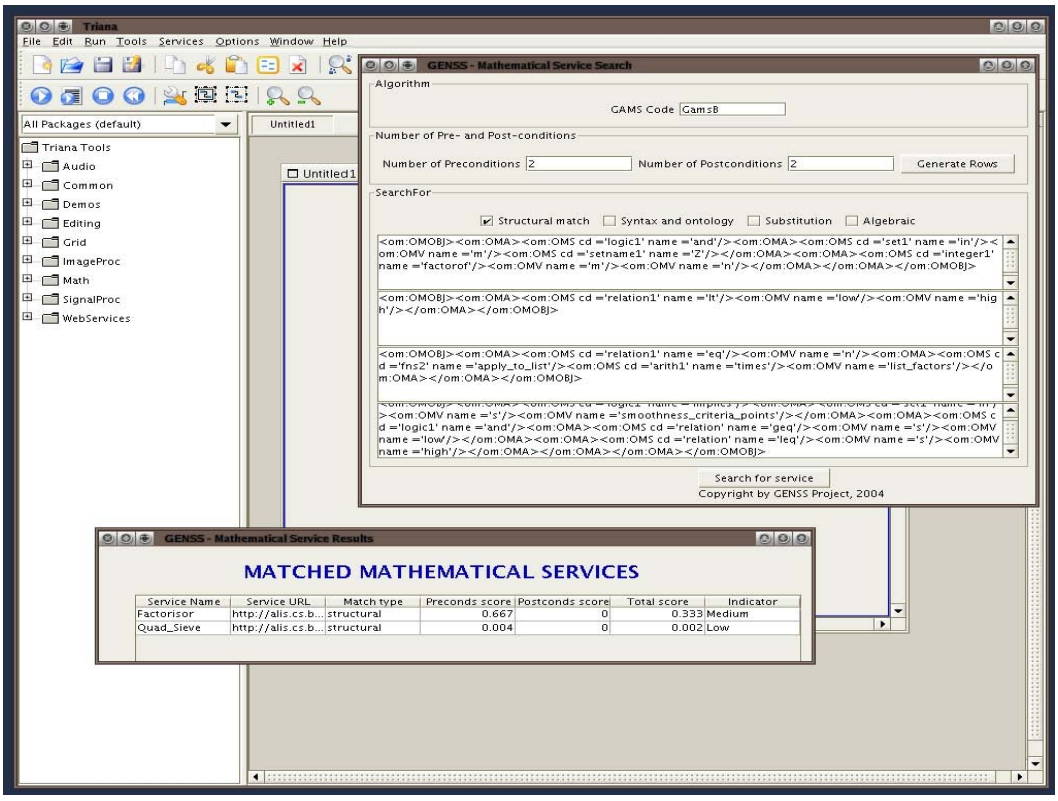


Figure 3: Results from the matchmaker in Triana

be implemented by the user to achieve this. This mode of use is particularly useful if a user does not intend to use the matchmaker within a workflow, but prefers to investigate services of interest, before a workflow is constructed graphically.

Using the matchmaker as a search service allows it to be used as a standard component in Triana. In this case, a user constructs a workflow with a search service being a component in this workflow. During enactment, the search service invokes the matchmaker and returns one or more results. Where a single result is returned (i.e. where the matchmaker is being used as a broker), the workflow continues to the next service in graph without any requirement for user input. As Triana already allows for dynamic binding of components to service instances, the search service essentially provides an alternative form of dynamic binding.

5 Complexity issues

An important issue for brokerage systems is whether the system scales well with respect to the number of services registered with the system. That is, the number of services should only have a small factor in any expression calculating the complexity of the system. For the brokerage system described in this paper, the complexity costs will be dependant on the complexity costs of the matchmaker plugins to the service. Generally the more powerful the plugin, the higher the complexity cost. One class of matchmaker plugins, that we utilise is ontology based plugins, these generally involve some traversal of an ontology and as such, the costs will be dependant more on the height than the absolute size of the ontology. Some figures indicating the actual performance of our system appear in Appendix A

6 Planned Future Work

A particular system which manages allocation of jobs to a pool of processors, is the GridSAM system [8]. The GridSAM system takes job descriptions given in JSDL [2] and the executables from Clients. It then distributes the jobs over the processors which perform the processing, GridSAM then returns the results to the Clients. It also provides certain monitoring services. Grimoires [9] is a registry service for services, which allows attachments of semantic descriptions to the services. Grimoires is UDDIv2 [1] compliant and stores the semantic attachments as RDF triples, this gives scope for attachments with arbitrary schema including those for describing mathematical services. A restriction that has been identified with the current GridSAM architecture, is that it doesn't incorporate any brokerage capabilities. Furthermore it appears that the Grimoires registry does not provide the resource allocation provided by GridSAM. A future project will look at integration of these two approaches in such a manner that it can be used in coordination with the architecture described in this paper.

7 Related Work

Matchmaking has quite a significant body of associated literature, so we do not attempt to be exhaustive, but survey just those systems that have been influential or we believe are especially relevant to the issues raised here, namely architecture, flexibility and matching technologies.

Although generalizations are risky, broad categorizations of matchmaking and brokerage research seem possible using criteria such as domain, reasoning mechanisms and adaptability.

Much of the published literature has described generic brokerage mechanisms using syntactic or semantic, or a combination of both, techniques. Some of the earliest systems, enabled by the development of KIF (Knowledge Interchange Format) [5] and KQML (Knowledge Query and Manipulation Language) [20], are SHADE [11] operating over logic-based and structured text languages and the complementary COINS [11] that operates over free text using well-known term-first index-first information retrieval techniques. Subsequent developments such as InfoSleuth [14] applied reasoning technology to the advertised syntax and semantics of a service description, while the RETSINA system [19] had its own specialized language [18] influenced by DAML-S (the pre-cursor to OWL-S) and used a belief-weighted associative network representation of the relationships between ontological concepts as a central element of the matching process. While technically sophisticated, a particular problem with the latter was how to make the initial assignment of weights without biasing the system inappropriately. A distinguishing feature of all these systems is their monolithic architecture, in sharp contrast to GRAPPA [22] (Generic Request Architecture for Passive Provider Agents) which allows for the use of multiple matchmaking mechanisms. Otherwise GRAPPA essentially employs fairly conventional multi-attribute clustering technology to reduce attribute vectors to a single value. Finally, a notable contribution is the MathBroker architecture, that like the domain-specific plug-ins of our brokerage scheme, works with semantic descriptions of mathematical services using the same MSDL language. However, current publications [3] seem to indicate that matching is limited to processing taxonomies and the functional issues raised by pre- and post-conditions are not considered. The MONET broker [4], in conjunction with the RACER reasoner and the Instance Store demonstrated one of the earliest uses of a description logic reasoner to identify services based on taxonomic descriptions coming closest to the objective of the plug-ins developed for GENSS in attempting to provide functional matching of task and capability.

In contrast, matching and brokerage in the grid computing

domain has been relatively unsophisticated, primarily using syntactic techniques, such as in the ClassAds system [16] used in the Condor system and RedLine [12] which extends ClassAds, where match criteria may be expressed as ranges and hence are a simple constraint language. In the Condor system, the use of ClassAds is to enable computational jobs find suitable resources, generally using dynamic attributes such as available physical and virtual memory, CPU type and speed, current load average, and other static attributes such as operating system type, job manager etc. A resource also has a simple policy associated with it, which identifies when it is willing to accept new job requests. The approach is therefore particular focused to work for managing job execution on a Condor pool, and configured for such a system only. It would be difficult to deploy this approach (without significant changes) within another job execution system, or one that makes use of a different resource model. The RedLine system allows matching of job requests with resource capabilities based on constraints – in particular the ability to also search based on resource policy (i.e. when a resource is able to accept new jobs, in addition to job and resource attributes). The RedLine description language provides functions such as *Forany* and *Forall* to be able to find multiple items that match. The RedLine system is however still constrained by the type of match mechanisms that it supports—provided through its description language. Similar to Condor, it is also very difficult to modify it for a different resource model. Our approach is more general, and can allow plug-ins to be provided for both RedLine and Condor as part of the matchmaker configuration. In our model, therefore, as the resource model is late-bound, we can specify a specialist resource model and allow multiple such models to co-exist, each implemented in a different configuration of the matchmaker.

8 Conclusion

We have outlined the development of a brokerage architecture whose initial requirements were derived from the MONET broker, namely the discovery of mathematical services, but with the addition of the need to establish a functional relationship between the pre- and post-conditions of the task and the capability. As a consequence of the engineering approach taken in building the GENSS matchmaker/broker, the outcome has been a generic matchmaking shell, that may be populated by a mixture of generic and domain-specific plug-ins. These plug-ins may also be composed and deployed with low overhead, especially with the help of workflow tools, to create bespoke matchmakers/brokers. The plug-ins may be implemented as web services and a mechanism has been provided to integrate them into the architecture. Likewise, the use of web services for the plug-ins imposes a low overhead on the production of new components which may thus encourage wider authorship of new, shareable, generic and specific matching elements (as reported in the Appendix). This would also provide the basis for defining a policy about how results from multiple matching techniques may be combined.

9 Acknowledgements

The work reported here is partially supported by the Engineering and Physical Sciences Research Council under the Semantic Grids call of the e-Science program (GENSS grant reference GR/S44723/01) and partially supported through the Open Middleware Infrastructure Institute managed program (project KNOOGLE).

References

[1] A.E. Walsh. UDDI, SOAP, and WSDL: The Web Services Specification Reference Book, 2002. UDDIORG.

[2] Stephen McGough Ali Anjomshoaa, Darren Pulsipher. Job Submission Description Language WG (JSDL-WG), 2003. Available from `https://forge.gridforum.org/projects/jsdl-wg/`.

[3] Rebhi Baraka, Olga Caprotti, and Wolfgang Schreiner. A Web Registry for Publishing and Discovering Mathematical Services. In *EEE*, pages 190–193. IEEE Computer Society, 2005.

[4] Olga Caprotti, Mike Dewar, and Daniele Turi. Mathematical Service Matching Using Description Logic and OWL. In Andrea Asperti, Grzegorz Bancerek, and Andrzej Trybulec, editors, *MKM*, volume 3119 of *Lecture Notes in Computer Science*, pages 73–87. Springer, 2004.

[5] M. Genesereth and R. Fikes. Knowledge Interchange Format, Version 3.0 Reference Manual. Technical report, Computer Science Department, Stanford University, 1992. Available from `http://www-ksl.stanford.edu/knowledge-sharing/papers/kif.ps`.

[6] C. Goble. Putting Semantics into e-Science and Grids. *Proc E-Science 2005, 1st IEEE Intl Conf on e-Science and Grid Technologies, Melbourne, Australia, 5-8 December*, 2005.

[7] EO 600 Gravitational Wave Project. `http://www.geo600.uni-hannover.de/`.

[8] GridSAM - Grid Job Submission and Monitoring Web Service, 2006. Available from `http://gridsam.sourceforge.net/2.0.0-SNAPSHOT/index.html`.

[9] (Grimoires: Grid RegIstry with Metadata Oriented Interface: Robustness, Efficiency, Security , 2004. Available from `http://twiki.grimoires.org/bin/view/Grimoires/`.

[10] Duncan Hull, Robert Stevens, Phillip Lord, Chris Wroe, and Carole Goble. Treating "shimantic web" syndrome with ontologies. In John Domingue, editor, *First Advanced Knowledge Technologies workshop on Semantic Web Services (AKT-SWS04)*, volume 122. KMi, The Open University, Milton Keynes, UK, 2004. Workshop proceedings available from CEUR-WS.org. ISSN:1613-0073. `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-122/`.

[11] D. Kuokka and L. Harada. Integrating information *via* matchmaking. *Intelligent Information Systems 6(2-3), pp. 261-279*, 1996.

[12] Chuang Liu and Ian T. Foster. A Constraint Language Approach to Matchmaking. In *RIDE*, pages 7–14. IEEE Computer Society, 2004.

[13] Simone Ludwig, Omer Rana, William Naylor, and Julian Padget. Matchmaking Framework for Mathematical Web Services. *Journal of Grid Computing*, 4(1):33–48, March 2006. Available via `http://dx.doi.org/10.1007/s10723-005-9019-z`. ISSN: 1570-7873 (Paper) 1572-9814 (Online).

[14] W. Bohrer M. Nodine and A.H. Ngu. Semantic brokering over dynamic heterogenous data sources in InfoSleuth. In *Proceedings of the 15th International Conference on Data Engineering, pp. 358-365*, 1999.

[15] Mathematics on the Net - MONET. `http://monet.nag.co.uk`.

[16] Rajesh Raman, Miron Livny, and Marvin H. Solomon. Matchmaking: Distributed Resource Management for High Throughput Computing. In *HPDC*, pages 140–, 1998.

[17] O. F. Rana, Ali Shaikh Ali, and Ian J. Taylor. Web Services Composition for Distributed Data Mining. In D. Katz, editor, *Proc. of ICPP, Workshop on Web and Grid Services for Scientific Data Analysis, Oslo, Norway June 14*, 2005.

[18] K. Sycara, S. Widoff, M. Klusch, and J. Lu. Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. *Journal of Autonomous Agents and Multi Agent Systems*, 5(2):173–203, June 2002.

[19] Katia P. Sycara, Massimo Paolucci, Martin Van Velsen, and Joseph A. Giampapa. The RETSINA MAS Infrastructure. *Autonomous Agents and Multi-Agent Systems*, 7(1-2):29–48, 2003.

[20] D. McKay T. Finin, R. Fritzson and R. McEntire. KQML as an agent communication language. In *Proceedings of 3rd International Conference on Information and Knowledge Management, pp. 456-463*, 1994.

[21] Ian Taylor, Matthew Shields, Ian Wang, and Roger Philp. Grid Enabling Applications using Triana. *Workshop on Grid Applications and Programming Tools. In conjunction with GGF8. Organized by: GGF Applications and Testbeds Research Group (APPS-RG) and GGF User Program Development Tools Research Group (UPDT-RG)*, 2003.

[22] D. Veit. *Matchmaking in Electronic Markets*, volume 2882 of *LNCS*. Springer, 2003. Hot Topics.

[23] Amy Moormann Zaremski and Jeannette M. Wing. Specification Matching of Software Components. *ACM Transactions on Software Engineering and Methodology*, 6(4):333–369, October 1997.

A GENSS performance indicators

Performance indicators for the GENSS matchmaker are reported here. An "end-to-end" test, with a user in Bath and a registry in Cardiff, produced a wall-clock time of 8 seconds. A "soak" test to repeatedly run the same search (client/matching process – as a Web Service – on a machine in Bath with registry in Cardiff) produced the following results:

Structural matcher

Iterations	real	user	sys
1	0m14.495s	0m4.362s	0m0.172s
10	1m2.898s	0m5.766s	0m0.318s
100	9m59.015s	0m12.173s	0m1.566s

Indicating that amortized elapsed time/query is around 6 seconds using the structural matcher.

Ontological matcher

Iterations	real	user	sys
1	0m22.089s	0m4.786s	0m0.173s
10	1m49.031s	0m5.882s	0m0.299s
100	17m13.894s	0m12.682s	0m1.203s

Indicating that amortized elapsed time/query is just over 10 seconds using the ontological matcher. Performing the same test as above, but where a command line Java application was used to search the service registry, linux system monitoring tools report that on the machine carrying out the search process that program used approximately 25.4MB and 36.06% CPU.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Best Practices in Web Service Style, Data Binding and Validation for use in Data-Centric Scientific Applications

Asif Akram, **David Meredith** and Rob Allan

e-Science Centre, CCLRC Daresbury Laboratory, UK
a.akram@dl.ac.uk, d.j.meredith@dl.ac.uk, r.j.allan@dl.ac.uk

Abstract

We provide a critical evaluation of the different Web Service styles and approaches to data-binding and validation for use in 'data-centric' scientific applications citing examples and recommendations based on our experiences. The current SOAP API's for Java are examined, including the Java API for XML-based remote procedure calls (JAX-RPC) and Document style messaging. We assess the advantages and disadvantages of 'loose' verses 'tight' data binding and outline some best practices for WSDL development. For the most part, we recommend the use of the document/ wrapped style with a 100% XML schema compliant data-model that can be separated from the WSDL definitions. We found that this encouraged collaboration between the different partners involved in the data model design process and assured interoperability. This also leverages the advanced capabilities of XML schema for precisely constraining complex scientific data when compared to RPC and SOAP encoding styles. We further recommend the use of external data binding and validation frameworks which provide greater functionality when compared to those in-built within a SOAP engine. By adhering to these best practices, we identified important variations in client and experimental data requirements across different institutions involved with a typical e-Science project.

1. Introduction

A Service Oriented Architecture is an architectural style whose goal is to achieve loose coupling among interacting software agents (services and clients). A SOA achieves loose coupling by employing two architectural constraints: 1) a small set of well-defined interfaces to all participating software agents and, 2) ensuring the interfaces are universally available to all providers and consumers. In simple terms, a service is a function that is self-contained and immune to the context or state of other services. These services can communicate with each other, either through explicit messages, or by a number of 'master' services that coordinate or aggregate activities together, typically in a workflow. In recent years, Web services have been established as a popular technology for implementing a SOA. The well-defined interface required for services is described in a WSDL (Web Service Description Language) file. Services exposed as Web services can be integrated into complex workflows which may, in a typical e-Science project, span multiple domains and organizations.

2. Binding/ Encoding Styles

The following section examines the different styles of WSDL file, and the resulting format of each SOAP message. It is important to understand that the WSDL binding style dictates the style of SOAP encoding (formatting) of the SOAP message that is transmitted 'over the wire.' This has serious implications upon Web Service interoperability. Collectively, the process of generating SOAP messages according to different styles of WSDL file is referred to as 'SOAP encoding' or 'WSDL binding,' and can either be Remote Procedure Call (RPC), or Document style. These two Web Service styles represent the RPC-centric and Message-centric view points. Most of the documentation however, focuses on the simpler RPC-centric viewpoint and often gives the misleading impression that SOAP and Web services are just another way of doing RPC. Table 1 provides a comparison of the different WSDL binding styles and resulting styles of SOAP encoding. The schema examples are referred to in the text in the following section. Table 2 provides a summary of the main advantages and disadvantages of each approach.

2.1 RPC Encoding/ Binding Style

The following summary examines the key features of the RPC WSDL binding style and the format of a resulting SOAP message. Table 1 illustrates these key points (schema examples are numbered and referred to in the text).

2.1.1 RPC (applies to encoded and literal)

- An RPC style WSDL file contains multiple <part> tags per <message> for each request/ response parameter (10b, 11b).
- Each message <part> tag defines type attributes, not element attributes (message parts are not wrapped by elements as in Document style WSDL files) (10b, 11b).
- The type attribute in each <part> tag can either; a) reference a complex or simple type defined in the <wsdl:types> section, e.g. <part name="x" type="tns:myType">, or b) define a simple type directly, e.g. <part name="x" type="xsd:int"> (10b, 11b respectively).
- An RPC SOAP request wraps the message parameters in an element named after the invoked operation (2a, 12a). This 'operational wrapper element' is defined in the WSDL target namespace. An RPC SOAP response wraps the message parameters in an element named after the invoked operation with 'Response' appended to the element name.
- The difference between RPC encoded and RPC literal styles relates to how the data in the SOAP message is serialised/ formatted when sent 'over the wire'. The abstract parts of the WSDL files are similar (i.e. the <wsdl:types>, <wsdl:message> and <wsdl:portType> elements – refer to Section 6.0). The only significant difference relates to the definition of the <wsdl:binding> element. The binding element dictates how the SOAP message is formatted and how complex data types are represented in the SOAP message.

2.1.2 RPC/ encoded

- An RPC/ encoded WSDL file specifies an encodingStyle attribute nested within the <wsdl:binding>. Although different encoding styles are legal, the most common is SOAP encoding. This encoding style is used to serialise data and complex types in the SOAP message. (http://schemas.xmlsoap.org/soap/encoding).
- The use attribute, which is nested within the <wsdl:binding> has the value "encoded".
- An RPC/ encoded SOAP message has type encoding information for each parameter element. This is overhead and degrades throughput performance (4a, 7a, 8a).
- Complex types are SOAP encoded and are referenced by "href" references using an identifier (3a). The href's refer to "multiref" elements positioned outside the operation wrapping element as direct children of the <soap:Body> (6a). This complicates the message as the <soap:Body> may contain multiple "multiref" elements.
- RPC/ encoded is not WS-I compliant [1] which recommends that the <soap:Body> should only contain a single nested sub element.

2.1.3 RPC/ literal

- RPC/ literal style improves upon the RPC/ encoded style.
- An RPC/ literal WSDL does not specify an encodingStyle attribute.
- The use attribute, which is nested within the <wsdl:binding>, has the value "literal".
- An RPC/literal encoded SOAP message has only one nested child element in the <soap:Body> (12a). This is because all parameter elements become wrapped within the operation element that is defined in the WSDL namespace.
- The type encoding information for each nested parameter element is removed (14a, 15a).
- RPC/ literal is WS-I compliant [1].

The main weakness with the RPC encoding style is its lack of support for the constraint of complex data, and lack of support for data validation. The RPC/ encoded style usually adopts the SOAP encoding specification to serialize complex objects which is far less comprehensive and functional when compared to standard XML Schema. Validation is also problematic in the RPC/ literal style; when an RPC/ literal style SOAP message is constructed, only the operational wrapper element remains fully qualified with the target namespace of the WSDL file, and all other type encoding information is removed from nested sub-elements (this is shown in Table 1). This means all parts/ elements in the SOAP message share the WSDL file namespace and lose their original Schema namespace definitions. Consequently, validation is only possible for limited scenarios, where original schema elements have the same namespace as the WSDL file. For the most part however, validation becomes difficult (if not impossible) since qualification of the operation name comes

from the WSDL definitions, not from the individual schema elements defined in the <wsdl:types> section.

2.2 Document Encoding/ Binding Style

In contrast to RPC, Document style encoding provides greater functionality for the validation of data by using standard XML schema as the encoding format for complex objects and data. The schema defined in the <wsdl:types> section can be embedded or imported (refer to Section 6.1). The following summary examines the key features of the Document WSDL binding style and the format of a resulting SOAP message. Table 1 illustrates these key points (schema examples are numbered and referred to in the text).

2.2.1 Document (applies to literal and wrapped)

- Document style Web services use standard XML schema for the serialisation of XML instance documents and complex data.
- Document style messages do not have type encoding information for any element (23a, 24a), and each element in the soap message is fully qualified by a Schema namespace by direct declaration (22a), or by inheritance from an outer element (30a).
- Document style services leverage the full capability of XML Schema for data validation.

2.2.2 Document/ literal

- Document/ literal messages send request and response parameters to and from operations as direct children of the <soap:Body> (22a, 26a).
- The <soap:Body> can therefore contain many immediate children sub elements (22a, 26a).
- A Document/literal style WSDL file may therefore contain multiple <part> tags per <message> (19b, 20b).
- Each <part> tag in a message can specify either a type or an element attribute, however, for WS-I compliance, it is recommended that only element attributes be defined in <part> tags for Document style WSDL (19b, 20b).
- This means that every simple or complex type parameter should be wrapped as an element and be defined in the <wsdl:types> section (15b, 16b).
- The main disadvantages of the Document/ literal Web Service style include: a) the operation name is removed from the <soap:Body> request which can cause interoperability problems (21a), and b) the <soap:Body> will contain multiple children (22a, 26a) if more than one message part is defined in a request/ response message (19b, 20b).
- Document/ literal is not fully WS-I compliant [1], which recommends that the <soap:Body> should only contain a single nested sub element.

2.2.3 Document/ wrapped

- An improvement on the Document/ literal style is the Document/ wrapped style.
- When writing this style of WSDL, the request and response parameters of a Web Service operation (simple types, complex types and elements) should be 'wrapped' within single all-encompassing request and response elements defined in the <wsdl:types> section (24b - akin to the RPC/ literal style).
- These 'wrapping' elements need to be added to the <wsdl:types> section of the WSDL file (24b).
- The request wrapper element (24b) must have the same name as the Web Service operation to be invoked (this ensures the operation name is always specified in the <soap:Body> request as the first nested element).
- By specifying single elements to wrap all of the request and response parameters, there is only ever a single <part> tag per <message> tag (32b).
- A Document/ literal style WSDL file is fully WS-I compliant [1] because there is only a single nested element in the <soap:Body> (29a).
- Document/ wrapped style messages are therefore very similar to RPC/ literal style messages since both styles produce a single nested element within a <soap:Body>. The only difference is that for Document/ wrapped style, each element is fully qualified with a Schema namespace.

The main advantage of the Document style over the RPC style is the abstraction/ separation of the type system into a 100% XML Schema compliant data model. In doing this, several important advantages related to data binding and validation are further realised which are discussed in the next section.

Table 1 - A Comparison of the Different WSDL Binding Styles and SOAP Encoding

Style	SOAP Request	WSDL
RPC Encoded	1a <soapenv:Body> 2a <getIndex xmlns:="urn:ehtpx-process"> 3a <admin href="#id0"/> 4a <URL xsi:type="xsd:string"> </URL> 5a </getIndex> 6a <multiRef id="id0" ……> 7a <email xsi:type=”xsd:string”> </email> 8a <PN xsi:type=”xsd:string”> </PN> 9a </multiRef> 10a </soapenv:Body>	1b <types> 2b <complexType name="AdminT"> 3b <sequence> 4b <element name="email" type="enc:string"/> 5b <element name="PN" type="enc:string"/> 6b </sequence> 7b </complexType> 8b </types> 9b <wsdl:message name="getIndexRequest"> 10b <wsdl:part name="admin" type="tns:AdminT"/> 11b <wsdl:part name="URL" type="enc:string"/> 12b </wsdl:message>
RPC Literal	11a <soapenv:Body> 12a <getIndex xmlns="urn:ehtpx-process"> 13a <admin xmlns=""> 14a <email> </email> 15a <PN> </PN> 16a </admin> 17a <URL xmlns=""> </URL> 18a </getIndex> 19a <soapenv:Body>	
Doc Literal	20a <soapenv:Body> 21a 22a <admin xmlns="urn:ehtpx-process"> 23a <email xmlns=""> </email> 24a <PN xmlns=""> </PN> 25a </admin> 26a <URL xmlns=""> </URL> 27a </soapenv:Body>	13b <types> 14b <complexType name="AdminT">… </complexType> 15b <element name=”admin” type=”tns:AdminT”> 16b <element name=”URL” type=” enc:string”> 17b </types> 18b <wsdl:message name="getIndexRequest"> 19b <wsdl:part name="in0" element="tns:admin"/> 20b <wsdl:part name="URL" element="enc:string"/> 21b </wsdl:message>
Doc Wrapped	28a <soapenv:Body> 29a <getIndex xmlns=" urn:ehtpx-process"> 30a <admin> 31a <email xmlns=""> </email> 32a <PN xmlns=""> </PN> 33a </admin> 34a <URL xmlns=""> </URL> 35a </getIndex> 36a </soapenv:Body>	22b <types> 23b <complexType name="AdminT"> ...</complexType> 24b <element name ="getIndex"> 25b <complexType> <sequence> 26b <element name ="admin" type =" tns:AdminT“/> 27b <element name =" URL " type =" xsd:string" /> 28b </sequence> </complexType > 29b </element> 30b </types> 31b <wsdl:message name="getIndexRequest"> 32b <wsdl:part name="in0" element="tns:getIndex"/> 33b </wsdl:message>

Table 2 - Advantages and Disadvantages of Each WSDL Binding Style and SOAP Encoding

Style	Advantages	Disadvantages
RPC Encoded	• Simple WSDL • Operation name wraps parameters	• Complex types are sent as multipart references meaning <soap:Body> can have multiple children • Not WS-I compliant • Not interoperable • Type encoding information generated in soap message • Messages can’t be validated • Child elements are not fully qualified
RPC Literal	• Simple WSDL • Operation name wraps parameters • <soap:Body> has only one element • No type encoding information • WS-I compliant	• Difficult to validate message • Sub elements of complex types are not fully qualified.
Doc Literal	• No type encoding information • Messages can be validated • WS-I compliant but with restrictions • Data can be modelled in separate schema	• WSDL file is more complicated • Operation name is missing in soap request which can create interoperability problems • <soap:Body> can have multiple children • WS-I recommends only one child in <soap:Body>
Doc Wrapped	• No type encoding information • Messages can be validated • <soap:Body> has only one element • Operation name wraps parameters • WS-I compliant	• WSDL file is complicated – request and response wrapper elements may have to be added to the <wsdl:types> if original schema element name is not suitable for Web Service operation name.

3. Data Abstraction, Data Binding and Validation

Abstraction of the Web Service type system into a 100% XML Schema compliant data model produces several important advantages;

- *Separation of Roles*
  The type system can be fully abstracted and developed in isolation from the network protocol and communication specific details of the WSDL file. In doing this, the focus becomes centred upon the business/ scientific requirements of the data model. In our experience, this greatly encourages collaboration between the scientists who are involved with the description of scientific data and data model design.
- *Data Model Re-usability*
  Existing XML Schema can be re-used rather than re-designing a new type system for each new Web Service. This helps reduce development efforts, cost and time.
- *Isolation of Changing Components*
  In our experience, the data model is the component that is most subject to change, often in response to changing scientific requirements. Its isolation therefore limits the impact on other Web Service components such as the concrete WSDL file implementation (see Section 6.0).
- *Avoid Dependency on SOAP Namespaces and Encoding Styles*
  Manual modeling of XML Schema may constitute extra effort but this gives the developer the most control and avoids using SOAP framework dependent namespaces and encoding styles. Most of the SOAP frameworks are traditionally RPC-centric and create WSDL based on the RPC/ encoding style which is not WS-I compliant. This also applies to languages other than Java.
- *Full XML Schema Functionality*
  The XML Schema type system leverages the more powerful features of the XML Schema language for description, constraint and validation of complex data (e.g. XSD patterns/ regular expressions, optional elements, enumerations, type restrictions etc). In our experience, this has proven invaluable for the description and constraint of complex scientific data.
- *Pluggable' Binding and Validation Frameworks*
  In most JAX-RPC implementations, XML serialization of a message's encoded parameters is hidden from the developer, who works with objects created automatically from XML data using semi-standardized mapping schemes for the generation of client and server stub/ skeleton classes. Consequently, the developer is both hidden from, and dependent upon the data binding/ validation framework of the SOAP engine. In our experience, SOAP engine data binding frameworks are usually not 100% Schema complaint, and often do not support the more advanced features of XML Schema (e.g. xsd:patterns). We believe that this is a major source of ambiguity and in our experience, this has often been a source of error that is beyond immediate control of the developer. An alternative approach is to use a dedicated, 100% Schema compliant data binding/ validation framework that is independent of the SOAP engine for the construction and validation of Web Service messages and instance documents (e.g. JAXB [2], XMLBeans [3]). Developers still manipulate XML in the familiar format of objects (courtesy of the binding framework), but there is no dependency upon the SOAP engine. In doing this, the more powerful/ functional features of an external binding framework can be levered, and the roles of the SOAP engine and data binding/validation framework become clearly separated into 'data-specific' and 'communication-specific' roles.
- *On-Demand Document Construction and Validation*
  This clear separation of roles means that XML messages/ documents can be constructed and validated at times when the SOAP engine is not required, for example, when constructing messages over an extended period of time (e.g. graphically through a GUI) and especially for the purposes of persistence (e.g. saving validated XML to a database). The separation of the data binding from the SOAP engine is gaining popularity in the next generation of SOAP engines that are now beginning to implement 'pluggable' data bindings (e.g. Axis2 [4] and Xfire [5]).

4. Loose Versus Strong Data Typing

A 'loosely typed' Web Service means the WSDL file does not contain an XML schema in

the type system to define the format of the messages, instead it uses generic data types to express communication messages. Loosely typed services are flexible, allowing Web Service components to be replaced in a 'plug-and-play' fashion. Conversely, a 'strongly typed' Web Service means the WSDL type system strictly dictates the format of the message data. Strongly typed Web services are less flexible, but more robust. Each style influences the chosen approach to data binding and each has its own advantages and disadvantages which are summarized in Table 3.

4.1 Loosely Typed Web services

A loosely typed WSDL interface specifies generic data types for an operation's input and output messages (either "String", "Base64-Encoded", "xsd:any", "xsd:anyType" or "Attachment" types). This approach requires extra negotiation between providers and consumers in order to agree on the format of the data that is expressed by the generic type. Consequently, an understanding of the WSDL alone is usually not sufficient to invoke the service.

- *"String" loose Data Type*

A String variable can be used to encode the actual content of a messages complex data. Consequently, the WSDL file may define simple String input/ output parameters for operations. The String input could be an XML fragment or multiple name value pairs (similar to Query string). In doing this, the implementation has to parse and extract the data from the String before processing. An XML document formatted as a String requires extra coding and decoding to escape XML special characters in the SOAP message which can drastically increase the message size.

- *"any"/ " anyType" loose Data Type*

The WSDL types section may define <xsd:any> or <xsd:anyType> elements. Consequently, any arbitrary XML can be embedded directly into the message which maps to a standard "javax.xml.soap.SOAPElement". Partners receive the actual XML but no contract is specified regarding what the XML data describes. Extraction of information requires an understanding of raw XML manipulation, for example, using the Java SAAJ API. Limited support for "any"/ "anyType" data type from various SOAP Frameworks may result in portability and interoperability issues.

- *"Base64 encoding" loose Data Type*

An XML document can be transmitted as a Base64 encoded string or as raw bytes in the body of a SOAP message. These are standard data types and thus every SOAP engine handles this data in compatible fashion. Indeed, embedding Base64 encoded data and raw bytes in the SOAP body is WS-I compliant.

- *"SOAP Attachment" loose Data Type*

SOAP attachments can be used to send data of any format that cannot be embedded within the SOAP body, such as raw binary files. Sending data in an attachment is efficient because the size of the SOAP body is minimized which enables faster processing (the SOAP message contains only a reference to the data and not the data itself). Additional advantages over other techniques include the ability to handle large documents, multiple attachments can be sent in a single Web Service invocation, and attachments can be compressed-decompressed for efficient network transport.

4.2 Strongly typed Web services

A purely strongly typed WSDL interface defines a complete definition of an operation's input and output messages with XML Schema, with additional constraints on the actual permitted values (i.e. Document style with value constraints). It is important to understand however, that Document style services do not have to be solely strongly typed, as they may combine both strongly typed fields with loose/ generic types where necessary. Strong typing is especially relevant for scientific applications which often require a tight control on message values, such as length of string values, range of numerical values, permitted sequences of values (e.g. organic compounds must have Carbon and Hydrogen in the chemical formula and rotation angle should be between 0 – 360 degrees).

From our experiences related to e-HTPX [6], the best approach involved mixing of the different styles where necessary. For mature Web services, where the required data is established and stable, the use of strong data typing was preferable. For immature Web services where the required data is subject to negotiation/ revision, loose typing was preferable. We often used the loose typing approach during initial developments and prototyping.

Table 3 – Advantages and Disadvantages of Loose versus Strong Data Typing in Web services

Modeling Approach	Advantages	Disadvantages
Loose Type	• Easy to develop • Easy to implement • Minimum changes in WSDL interface • Stable WSDL interface • Flexibility in implementation • Single Web Service implementation may handle multiple types of message • Can be used as Gateway Service routing to actual services based on contents of message	• Requires additional/ manual negotiation between client and service provider to establish the format of data wrapped or expressed in a generic type • This may cause problems regarding maintaining consistent implementations and for client/ service relations • No control on the messages • Prone to message related exceptions due to inconsistencies between the format of sent data, and accepted data format (requires Web Service code to be liberal in what it accepts – this adds extra coding complexity). • Encoding of XML as a string increases the message size due to escaped characters
Strong Type	• Properly defined interfaces • Tight control on the data with value constraints • Message validation • Different possibilities for data validation (pluggable data binding/validation) • Robust (only highly constrained data enters Web Service) • Minimized network overhead • Benefits from richness of XML	• Difficult to develop (requires a working knowledge of XML and WSDL) • Resistive to change in data model

5. Code First or WSDL First

For platform independence and Web Service interoperability, the WSDL interface should not reference or have dependencies upon any technical API other than XML Schema. However, the complexity of XML schema and over-verbosity of WSDL is a major concern in practical development of Web services. As a result, two divergent practices for developing Web services and WSDL have emerged, the ‘code first’ approach (also known as ‘bottom up’) and 'WSDL first' approach (also know as ‘top down’ or ‘contract driven’). The code first approach, which is often implemented in JAX-RPC environments, involves auto-generation of the WSDL file from service implementation classes using tools that leverage reflection and introspection. Alternatively, the WSDL first approach involves writing the original WSDL and XML Schema, and generating service implementation classes from the WSDL file.

5.1 Code First

Advantages

The 'code first' approach is often appealing because of its simplicity. Developers are hidden from the technical details of writing XML and WSDL.

Disadvantages

Generating WSDL files from source code often introduces dependencies upon the implementation language. This is especially apparent when relying upon the SOAP engine to serialize objects into XML, which can lead to interoperability issues across different platforms (e.g. differences in how Java and .NET serialize types that may be value types in one language but are reference objects in the other). WSDL created from source code is less strongly typed than WSDL that is created from the original XML Schema. Indeed, the more powerful features of XML Schema are often not supported by automatic WSDL generators.

5.2 WSDL First

Advantages

Platform and language interoperability issues are prevented, because both the client and server are working from a common set of interoperable XML Schema types. Defining a common platform-independent type system also facilitates separation of roles, whereby client side developers can work in isolation from server side developers. In our experience, this greatly increases productivity and simplifies development, especially for large distributed applications where developers may be geographically separated.

Disadvantages

The developer requires at least a reasonable knowledge of XML Schema and of WSDL.

In our experience, the WSDL first approach is the most suitable for developing robust, interoperable services. However, we also found the code first approach convenient for rapid

prototyping, especially when using loose data typing.

6. WSDL Modularisation

The WSDL specification and the WS-I basic profile recommend the separation of WSDL files into distinct modular components in order to improve re-usability and manageability. These modular components include;

1. XML Schema files.
2. An Abstract WSDL file.
3. A Concrete WSDL file.

6.1 XML Schema Files

Moving the type declarations of a Web Service into their own documents is recommended as data can be modeled in different documents according to namespace requirements. XML Schema declares two elements; <xsd:include> and <xsd:import> which are both valid children of the <wsdl:types> element (<xsd:include> is used when two schema files have the same namespace and <xsd:import> is used to combine schema files from different namespaces). In doing this, complex data types can be created by combining existing documents.

6.2 Abstract WSDL File

The abstract.wsdl file defines what the Web Service does by defining the data types and business operations of the Web Service. The file imports XML schema(s) as immediate children of the <wsdl:types> element, and defines different <wsdl:message> and <wsdl:portType> elements.

6.3 Concrete WSDL File

The concrete.wsdl file defines how and where to invoke a service by defining network protocol and service endpoint location with the <wsdl:binding> and <wsdl:service> elements. The concrete.wsdl file incorporates the abstract.wsdl file using <wsdl:import> or <wsdl:include>. These elements should be the first immediate children of the <wsdl:definitions> element (<wsdl:include> is used when two wsdl files have the same namespace and <wsdl:import> is used to combine wsdl files from different namespaces). This approach greatly improves component re-usability as the same abstract.wsdl file can have multiple service bindings.

7. Conclusions

Developments in the field of Web services have largely focused on the RPC style rather than on Document style messaging. This is apparent in the tools provided by vendors of JAX-RPC/ SOAP engine implementations. RPC style services have serious limitations for the description of data and can lead to interoperability issues. Real applications also require complex data modeling and validation support. For these applications, RPC simple typing and SOAP encoded complex types are inadequate. The RPC style can produce interoperability issues as many automatic WSDL generation tools introduce technical dependencies upon implementation languages. As a result, the RPC style is increasingly being referred to as ‘CORBA with brackets’ in the Web Service community. Loosely typed RPC services provide an alternative approach by encapsulating data within generic types. Loosely typed services are easy and convenient to develop and are suitable in a number of scenarios. However, loose typing introduces a different set of limitations, mainly associated with the additional manual negotiation between consumer and provider to establish the format of encapsulated data. In contrast to RPC, Document style services use 100% standard XML schema as the type system. This facilitates complex data modeling, loose and tight data typing where necessary, and full validation support. Use of a platform independent type system also ensures transport agnosticity, where abstract WSDL definitions can be bound to different transport protocols defined in concrete WSDL files. We also recommend the use of dedicated data binding/ validation frameworks for the construction of XML documents/ messages rather than relying on the SOAP engine. In doing this, the more powerful features of XML Schema can be levered, and the roles of the SOAP engine and data binding/ validation framework become clearly separated. The main disadvantage of the Document style is its increased complexity over RPC; developers require at least a reasonable understanding of XML and WSDL and are required to take the ‘WSDL first’ approach to Web Service design.

References / Resources

[1] WSI; http://www.ws-i.org/
[2] JAXB; http://java.sun.com/webservices/jaxb/
[3] XML Beans; http://xmlbeans.apache.org/
[4] Axis; http://ws.apache.org/axis/
[5] XFire; http://xfire.codehaus.org/
[6] The e-HTPX project; http://www.e-htpx.ac.uk

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Service-enabling Legacy Applications for the GENIE Project

Sofia Panagiotidi, Jeremy Cohen, John Darlington, Marko Krznarić and Eleftheria Katsiri

London e-Science Centre, Imperial College London, South Kensington Campus, London SW7 2AZ, UK
Email: {sp1003,jhc02,jd,marko,ek}@doc.ic.ac.uk

Abstract. We present work done within the Grid ENabled Integrated Earth system model (GENIE) project to take the original, complex, tightly-coupled Fortran earth modeling application that has been developed by the GENIE team and enable it for execution within a component-based execution environment. Further we have aimed to show that by representing the application as a set of high-level Java Web Service components, execution and management of the application can be made much more flexible. We show how the application has been built into higher-level components and how these have been wrapped within the Java Web Service abstraction. We then look at how these components can be composed into workflows and executed within a Grid environment.

Keywords: component, (earth) module, instance, port, grid, wrapper, glue (code), abstract interface, concrete interface

1 Introduction

Modularity and component wise construction is a central feature of all grid systems. Grids provide simplified execution of large-scale scientific applications, across multiple computational resources and organisations. The Grid ENabled Integrated Earth system model (GENIE) is a climate model simulation developed under the NERC-funded GENIE and GENIEfy projects. This computationally intensive application, written in Fortran, is ideally suited to Grid environments. However the existing, tightly-coupled, sequential application model cannot take advantage of the benefits of Grid systems.

We present work undertaken to modify the GENIE model allowing the existing Fortran implementation to take advantage of Grid execution features. GENIE consists of a set of modules simulating various earth entities (e.g. atmosphere, sea-ice etc.). Entities may have multiple module implementations that are coupled to produce a complete representation of an earth simulation. We present an abstract component model used to wrap the existing GENIE modules into higher-level components, taking into account features of the existing framework that present difficulties for traditional component models. Rather than rewriting the application from scratch in a higher-level language such as Java, an unreasonably complex task, we wrap the existing code in Java wrappers and show how the resulting components can be composed into workflows that may be executed through Grid middlewares.

The rest of the paper is as follows: section 2 discusses in more detail the structure of the GENIE application and some basic features of its architecture and implementation. Some previous work is summarised. Section 3 provides the neccessary background on grid-enabled component architectures. In the following section our visionary model for GENIE is presented and explained in detail. In section 5 some issues and technical details on how we have been working towards wrapping up GENIE modules in order to move closer to the goal are discussed. Last, Section 6 concludes and discusses further work.

2 Earth System Models and GENIE

2.1 General Description

GENIE [2] is a scalable modular platform aiming to simulate the long term evolution of the Earth's climate. Building an Earth system model (ESM) involves the coupling of a set of specialised components. Thus, distinct earth entities such as the atmosphere, the ocean, the

land, the sea-ice etc., referred to from now on as *earth modules* or just *modules*, are modelled independantly, using specified meshes of data points to represent the boundary surfaces, and conform to the natural laws of physics that govern the exchange of energy and fluxes from one to another.

One of the architectural characteristics of GENIE is that it is designed so as to consist of more than one implementation (*instance*) of an earth element in the case of the atmosphere, ocean, land and sea-ice. Furthermore, scientists are able to choose to experiment with several simulation scenarios (*configurations*) comprising of combinations of such implementations. The way this was implemented is through a separate program, "genie.F", which is responsible for the control and execution of each module, the attribute passing between the chosen ones and the appropriate interpolations of the data exchanged, through interpolation functions. In fact, all possible cases of configurations are handled through the "genie.F" with the use of flags, being switched on/off to specify the use of which implementation of a module.

As new modules are actively being researched and developed, it is desirable for the GENIE community to have the flexibility to easily add, modify and couple together GENIE modules and experiment with new configurations, without undue programming effort. Despite significant progress in the GENIE community, the desired result is far from reality.

2.2 Previous work

Several milestones towards the advance of the GENIE framework have been achieved. The first task was to separate the tightly coupled Fortran code into distinct pieces each representing an environmental module (atmosphere, land, ocean, ocean-surface, chemistry, sea-ice, etc). The main piece of code handling and coupling the modules was disengaged and formed the so-called "genie.F" top program.

Efforts in gradually moving GENIE towards a grid environment have been made in the past, most importantly allowing the execution and monitoring of many ensemble experiments in parallel [4, 6], reducing the overall execution time. Though, they all deal with a subset of the gridification issues and serve the isolated current at that time needs of the scientific community.

Previous work also includes research into the way the ICENI [3] framework can be used to run parameter sweep experiments across multiple Grid resources [5].

Lastly, there have been efforts into wrapping up each of the basic Earth modules using the JNI [7] library, which led to unexpected complications and unjustified amounts of effort, leading us to look into a more efficient solution.

3 Component-based Frameworks

The basic reason why the design of GENIE has not achieved desired level of modularity, which is the objective, is that it is fundamentally restricted by the use of Fortan as a scientifically efficient language [8].

The component programming model is the latest stage of a natural progression, starting from the monolithic application, and moving towards applications built from increasingly more modularised pieces, with one addition, the coupling framework. Components are designed to be protected from changes in the software environment outside their boundaries.

Thus, a component can be described as a 'black box', fully specified by its function specification and the input/output type patterns of the data (ports). The fact that a component is interfaced with other modules/systems only through its input/output ports (and is otherwise shielded from the rest of the world) allows bigger applications to be designed, implemented, and tested, independently of everything else.

The coupling framework within component architecture is a system defining the bindings between the components in a clear and complete way and providing information about the run-time environment. It can include connectors which perform the direct "binding" of the ports of two components or even more composite operations on the input/output/inout ports. Finally, a configuration of components may be (re)used as an individual component in another (sub) framework.

In this context, an application like GENIE may be composed (assembled) at run-time from components selected from a component pool. A component-based approach to GENIE would enable a user to fully exploit parallel execution of the system. Furthermore, by having seperate layers of software development as such, the configuration and desired behaviour of the simula-

tion model can be easily defined and modified by the user.

One of the desires of the GENIE community has been to experiment with the order of execution of the earth modules. With the current system, this is hard to achieve. Our component-based approach allows to easily modify the execution order of the components and the starting order of the components giving a possibility to a user to further test the robustness of the simulation method.

An example of an ideal composition environment would be the Imperial College e-Science Networked Infrastructure (ICENI) [3]. ICENI is a pioneering Service Oriented Architecture (SOA) and component framework developed at the London e-Science Centre, Imperial College London. Within the ICENI model an abstract component can have several differing implementations. Associated with each component is metadata describing its characteristics. At deployment, this metadata is used to achieve the optimal selection of component implementation and execution resource in order to minimise overall execution time given the Grid resources currently available.

4 A Grid Model for GENIE

4.1 Abstract Component Model

Within the ICENI model, semantically equivalent components have the same interface. In fact there need only be one semantically equivalent *abstract* component, concrete implementations inherit from this. In earth modelling systems such as GENIE, however, different implementations of a semantically equivalent earth entity module may have differing interfaces. For example, one ocean model may incorporate salinity while another does not. However, they are both ocean models and should be interchangable. We therefore extend the ICENI philosophy to provide different mechanisms whereby similar models but with differing interfaces can be used in an interchangable fashion. We do this by extending the notion of an interface by wrapping the module with plugin adapters to accomodate the different interfaces. We describe this interface model and show how the implementation can be achieved through Babel [1].

4.2 Extended Abstract Component Model

To realise this model we define four key questions which we then look at in greater detail:

- How are abstract models linked/composed?
- How is a GENIE system comprising various models assembled?
- How is GENIE system deployed in a distributed dynamic environment?
- How do GENIE models communicate with each other?

In order to compose abstract models in an interchangeable manner, it is necessary for the models to have common interfaces. In the case of GENIE, it is possible to encounter situations where semantically equivalent modules, that should in theory be interchangeable, cannot be substituted due to differing interfaces. We tackle this issue by defining a pluggable abstract interface that allows the knowledge of the module developer to be encapsulated within a 'plugin'.

To build a GENIE system from a given set of modules it is necessary to compose the modules in a semantically valid format. Using component metadata, an idea utilised extensively in ICENI components, we can annotate the GENIE components with information that determines their composability.

By wrapping GENIE components in a Java Web Service wrapper, it is possible to deploy the components in a service container allowing them to communicate using standard Web Service protocols. The use of Web Service standards provides mobility and allows the location of component deployment to be determined at application run time. The deployment of GENIE in a distributed, dynamic environment is discussed in more detail in section 5.

The communication between GENIE models may be accomplished using a variety of methods dependent on where they are deployed. Components deployed on geographically distributed resources may communicate using Simple Object Access Protocol (SOAP) messages over an HTTP connection, although this may not be the most efficient. Components that are co-located within a cluster may use MPI or shared memory to communicate (figure 1).

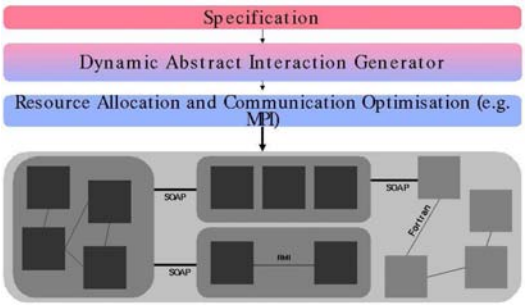


Fig. 1. Grid-based efficient model coupling and deployment

We now describe in more detail the idea of an abstract component that provides a common interface to a set of concrete implementations of a given type of model, even though those implementations may have differing interfaces. Figure 2 shows this model at the simplest level.

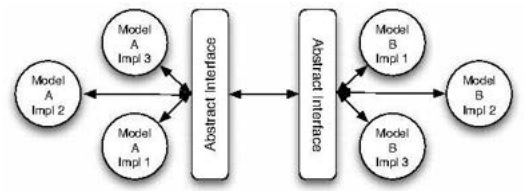


Fig. 2. A model using abstract interfaces to access concrete implementations.

The implementation of this abstract interface uses the idea of transformation plugins. A transformation plugin is a unit of code that is developed by the model developer and encapsulates their knowledge of the input and output format of data that their model accepts or produces. The plugin acts as a translation layer between a more general – although still model specific – interface and the concrete interface provided by the specific model implementation.

Due to the plugin architecture, we consider our abstract component interface to be more of a wrapper for a set of model implementations rather than simply a standard interface. This leads to the more detailed layout shown in figure 3. Our model supports both static, compile time generation of the abstract wrapper for a set of models and also dynamic, runtime registration of new models into the wrapper.

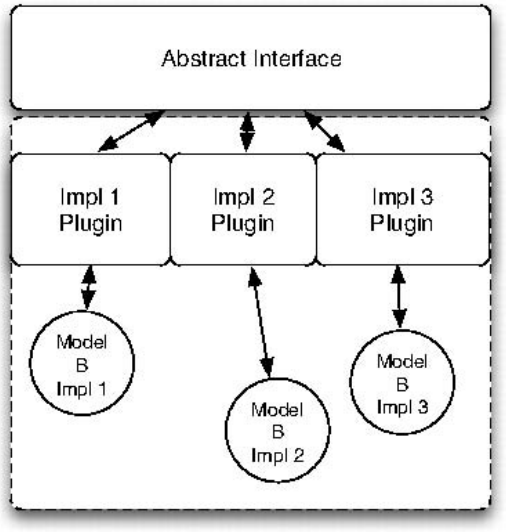


Fig. 3. Abstract interface marshalling requests to correct concrete implementation through plugin translators.

5 Grid Enabling the GENIE Application

5.1 Determining Component Ports

In the GENIE structure, each earth module is represented by a set of files, including Fortran routines, namelists, input files, netCDF data files and others. All these elements are represneted in the GENIE structure separately for each earth entity, to maintain the modularity of the code. The main functionality of each earth module is implemented in one subroutine. This subroutine requires/passes arguments corresponding to the physical quantities that the boundaries of the module receive and process. Furthermore, subroutine files containing initialisation, restart and finalising processes for the module are also part of its code structure. Thus, each physical module has its own hierarchy with a basic routine as well as initialisation/restart/finalisation routines. These routines are considered to be "top-level" as they are the only ones that directly communicate with the GENIE environment through the "genie.F" application controller.

In order to describe and access an earth module through an interface of a high level language, identifying the inputs and outputs is essential. This task, in the case of GENIE, is far from trivial. Although, as mentioned before, the arguments of the top-level routines are those that

need to be part of the interface, the separation of these arguments into inputs, outputs and inouts is difficult. The reason for this is that Fortran passes all arguments to a routine by reference. This means that it is very difficult to extract any information about whether a parameter is strictly used as read-only part of the memory, or gets modified during the execution of the specific piece of code.

During our inspection and detailed analysis of the GENIE modules, a long and complex task, the nature and type of the module parameters was documented. This served not only for our purposes, but has provided useful input to the GENIE project for future needs of the scientists and users of the application. In several cases, a manual exhaustive in-depth analysis of the code of a routine, together with the subroutines called within the code needed to be done in order to determine whether an attribute is written to, is of read-only nature, or possibly serves both read/modify purposes.

5.2 Linking Fortran to Java

In order to form high level components that can be launched and used as web services we choose Java to be the main language in which the interfaces of the earth modules will be exposed.

Rewriting them would possibly cause a number of inaccuracies and most importantly would cause a disruption in the development of such a complex scientific application. Therefore, we choose to wrap each earth module individually using an efficient and suitable mechanism.

For this purpose we pick Babel [1], an interoperability tool which addresses the language interoperability problem in software engineering. Babel uses a prototype SIDL (Scientific Interface Definition Language) interface description which describes a calling interface (but not the implementation) of a particular scientific module. Babel, as a parser tool, uses this interface description to generate glue code (server skeleton and client stub) that allows a software library implemented in one supported language to be called from any other supported language.

Below we denote some of the advantages when using Babel, as against other mechanisms, to wrap up GENIE modules:

- Babel provides all the three kinds of communication ports used by the GENIE subroutines – inputs, outputs, as well as inouts, the latter of which requires extra programming effort to be implemented via other mechanisms (e.g. JNI),
- It minimizes the problems we identified in [8] when using JNI, such as arrays stored differently in memory (C/Fortran), floating point numbers, complex structure argument passing, manual creation of intermediate ".h" files and programming effort.
- Possible future addition of extra components becomes easy, since it provides a standard mechanism for wrapping up components.
- Babel is compatible with most Fortran compilers used by scientists in GENIE.
- An SIDL file can include methods such as the initialise component, main and finalise, making it possible to expose all separately developed subroutines as parts of the same component, as semantically correct and desired, conforming to the component lifecycle of modern component architectures.
- An interoperability tool like Babel is particularly useful to make heterogeneous environments communicate, since our purpose is not only to make GENIE modules available over the web but moreover, to provide a generic methodology independent of the component implementation language.

After testing and verifying its suitability for the project's needs, we have been using Babel to wrap a sample of components and make them accessible from a Java client which is designed to replace "genie.F". Throughout this procedure several complications were encountered but resolved.

Firstly, Babel uses its own defined data types (from here and on referred to as *sidl-types*) in the SIDL interface specification that it accepts, producing code which handles similar data types in the language specified by the user. Thus, the code which is generated in Fortran, designed to access an earth module, needs some additions in order to serve its purpose. For this reason, several external routines "casting" the sidl-types of data from/to Fortran data types were developed. In this way, it was made possible to have a one-to-one correspondence of integers, floating point numbers, as well as one and two dimensional arrays to and from the equivalent sidl-types. Having these casting routines, the glue code can access the top level module routine by passing the arguments and having them re-

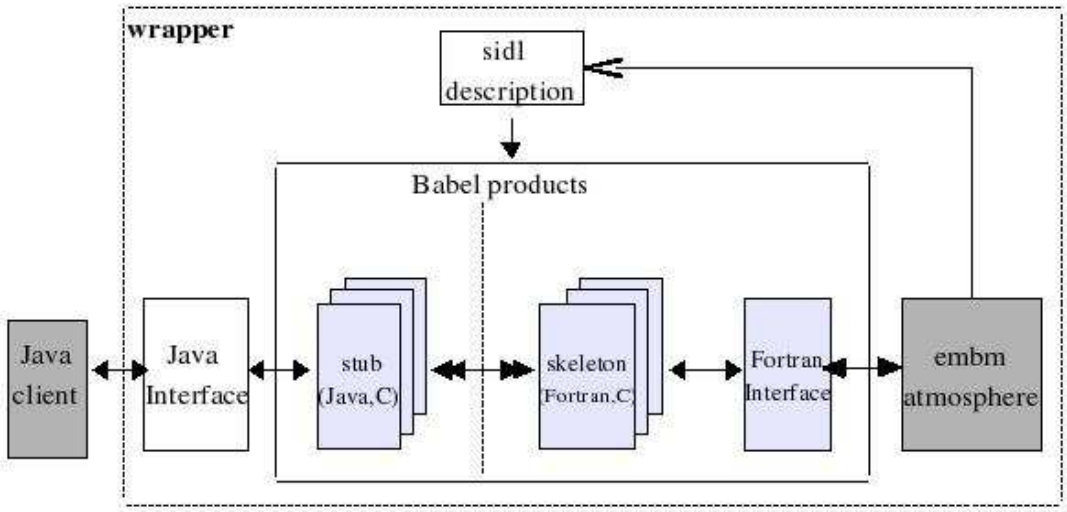


Fig. 4. Wrapping in Babel

turned after the addition of a piece of "casting" code inside the glue. It needs to be mentioned here that in this way, there is absolutely no need to modify any of the existing module code in order to access it from the skeleton created from Babel. Therefore, just by adding an additional structure for each module in the tree code of GENIE containing the SIDL descriptors and the generated glue code, the maintenance of the structure is guaranteed. Furthermore, this does not interfere with any attempt to run the GENIE application without using the wrapped modules and the componentised version, but executing it in the old-fashioned way where "genie.F" handles the modules.

Another feature of Babel is that it works with the use of shared libraries. The code of the module gets compiled together with the glue code to form a library, which then can be placed independently and accessed through a client from anywhere on the web. The GENIE application structure already contains the code of a module in an independent, shared library. Therefore, it is only sufficient to compile to glue code against the module library to ensure the communication of the two, allowing a more distributed environment where the skeleton accesses the actual implementation remotely.

We end up with a module wrapping procedure (figure 4) that enables an earth module (and eventually a whole configuration) to be accessed by a Java client, in the following simple stages, which require minimal programming effort:

- describe the input/output/inout ports of the module top level Fortran subroutine in a simple SIDL file
- run Babel parser to create glue code
- connect the top level subroutine of an earth module to the skeleton by passing the ports to/from Babel skeleton and to/from subroutine

5.3 Executing GENIE in a Grid Environment

So far, we have managed to wrap up two simple GENIE components using Babel and tested the results returned when calling them from a simple Java client. Our current activities mainly include a specific configuration (figure 5) comprising the GENIE basic ocean (Goldstein), a simple atmosphere (Embm) and a simple compatible sea-ice (Goldstein sea-ice). We have isolated this case as the most suitable to study and implement using Web Services. The three earth modules, together with their initialisation, and finalisation routines are being described and accessed through a Java interface and at the same time a Java client is being developed in order to access, and couple the components. Initially the configuration chosen is being tested by direct access of the interfaces exposed by the wrapped modules, without the use of Web Services, for reasons of simplicity.

It needs to be made clear that any component composition environment requires every entity to be in the form of a software component. Thus, all the functionality of "genie.F" must be disguised in a component's structure and become part of the data workflow. For example, all intermediate interpolation functions of data exchanged between modules as well as various functional and non-functional operators being part of the component composition phase, should become part of the (GENIE) application specification phase.

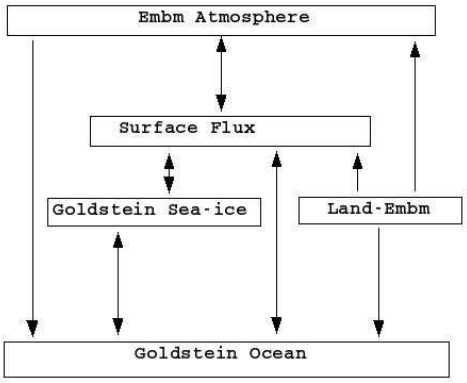


Fig. 5. Configuration to be implemented in Web Services

Below, we summarise the various enginnering steps which are planned to take place in order to achieve the grid enabling of the GENIE application using a Web Service-based workflow within the London e-Science Centre.

1. Wrap up a component in Java using Babel and test it individually to ensure proper results during execution.
2. Wrap up all modules for one existing configuration. Then create a Java client to synchronise the components and compare the results to those produced when executing "genie.F" over the same configuration.
3. Launch Web Services in place of the original components and modify the Java client to call these Web Services instead of the components.
4. Create all possible configurations of GENIE in a similar manner to that shown in the previous steps.
5. Create the Abstract Component Wrapper by choosing carefully what the concrete interface will be.
6. Use a workflow engine to orchestrate the Abstract Components in all configurations.
7. Advance the previous stages into a Grid environment so as to incorporate Coordination Specification, Dynamic Abstract Code Generation, Resource Allocation and Optimisation.

Although no thorough experiments over the efficiency have taken place yet, we expect that the advantages of this approach far outweight overhead produced during the execution (conversion of SIDL to language specific types and vice versa, data exchange between modules, data serialisation and trasportation over the Grid).

6 Conclusions

We have taken the Fortran implementation of the GENIE earth simulation application and shown how this can be modified to take advantage of a service-based computational Grid execution model. Through analysis of the original, tightly-coupled implementation, the component interfaces were identified. It was then possible to apply an abstract component model to the various earth entity module implementations to provide substitutable components that can be composed into workflows for scheduling and execution on Grid resources. Our implementation takes the form of a Service Oriented Architecture utilising Web Services as the service model. We have shown that it is possible to take a complex legacy application and apply wrapping techniques for service-enabling without the extensive effort that would be required for a rewrite of the original code in a different language. The work provides simplified deployment of GENIE workflows across multiple resources in combination with the opportunity for improved runtimes that distributed Grid execution allows. The generic and component-based nature of this architecture allows it to be applied into similar legacy system applications. It provides a language-independant environment where the rise of newly developed earth components can be easily incorporated to the rest of the application. Our showcase also describes generic and distinct stages (wrapping, deploying as web service, abstractly interfacing, interaction specification) in order to advance a legacy application into a loosely coupled, component-based application. We intend to continue this work to service enable further modules within the GENIE framework.

7 Acknowledgements

We would like to thank NERC who have funded the GENIE project and its follow up GENIEfy, Tim Lenton and the rest of the GENIE/GENIEfy project team.

References

1. Babel home page. http://www.llnl.gov/casc/components/babel.html.
2. Grid enabled integrated earth system model (genie). http://www.genie.ac.uk.
3. The imperial college e-science networked infrastructure (iceni). available at: http://www.lesc.ic.ac.uk/iceni.
4. M. Y. Gulamali, A. S. McGough, S.J. Newhouse, and J. Darlington. Using iceni to run parameter sweep applications across multiple grid resources. In *Global Grid Forum 10, Case Studies on Grid Applications Workshop*, Berlin, Germany, Mar. 2004.
5. M.Y. Gulamali, T.M. Lenton, A. Yool, A.R. Price, R.J. Marsh, N.R. Edwards, P.J. Valdes, J.L. Wason, S.J. Cox, M. Krznaric, S.J. Newhouse, and J. Darlington. Genie: Delivering e-science to the environmental scientist. In *UK e-Science All Hands Meeting 2003*, pages 145–152, Nottingham, UK, 2003.
6. M.Y. Gulamali, A.S. McGough, R.J. Marsh, N.R. Edwards, T.M. Lenton, P.J. Valdes, S.J. Cox, S.J. and Newhouse, and J. Darlington. Performance guided scheduling in genie through iceni. In *UK e-Science All Hands Meeting 2004*, pages 792–799, Nottingham, UK, 2004. ISBN=1-904425-21-6.
7. Sheng Liang. *Java Native Interface: Programmer's Guide and Specification*. Addison-Wesley, June 1999. ISBN=0201325772.
8. S. Panagiotidi, E. Katsiri, and J. Darlington. On advanced scientific understanding, model componentisation and coupling in genie. In *All Hands Meeting, Nottingham*, September 2005.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Grid computing in High Energy Physics using LCG: the BaBar experience

James Cunha Werner

University of Manchester

Abstract

This paper presents accounts of several real case applications using grid farms with data- and functional-parallelism. For data parallelism a job submission system (EasyGrid) provided an intermediate layer between grid and user software. For functional parallelism a PVM algorithm ran user's software in parallel as a master / slave implementation. These approaches were applied to typical particle physics applications: hadronic tau decays, searching for anti-deuterons, and neutral pion discrimination using genetic programming algorithms. Studies of performance show considerable reduction of time execution using functional gridification.

1. INTRODUCTION

The GridPP collaboration [1][2] and several other e-science projects have provide a distributed infrastructure and software middleware in UK. The LCG (Large Hadrons Collider Computing Grid) [3][4][5] software, developed by an international collaboration centered at CERN, provides a system for batch processing for High Energy Physics (HEP) through hundreds of computers connected by the Internet. It can be seen as a homogeneous common ground in a heterogeneous platform.

The **testbed farm** available at University of Manchester [6] contains 1 resource broker, 1 information system, 1 storage element (SE) with 7GB, 1 computer element (CE), and 6 worker nodes. The **production farm** contains 1 CE and 28 WNs, sharing other systems with the testbed. It is used to run analysis software from BaBar data.

The BaBar experiment [7] studies the differences between matter and antimatter, to throw light on the problem, posed by Sakharov, of how the matter-antimatter symmetric Big Bang can have given rise to today's matter-dominated universe.

This paper describes the approach for data gridification using EasyGrid (section 2), and for functional gridification (section 3) using real HEP analysis cases as benchmarks, followed by the conclusions.

2. Data gridification.

HEP Data Analysis can be divided in several hundreds of independent jobs running the same binary code in parallel over each dataset's data file with thousands of million of events.

Data gridification is implemented through EasyGrid Job Submission [6] software. It is an intermediate layer between Grid middleware and user's software. It integrates data, parameters, software, and grid middleware doing all submission and management of several users' software copies to grid.

EasyGrid's commands are: a. easymoncar: run Monte Carlo Events generation. b. easysub: run Raw data analysis. c. easygftp: run Generic data access applications using gridftp. d. easyapp: run Generic applications performing data gridification. e. easyroot: run Root application performing data gridification. f. easygrid: perform jobs' follow up, recover results in user's directory, and recover crash information for further analysis.

EasyGrid's first task is to find what event files are in the dataset (bookkeeper system), and what WNs have access to them (LFC metadata catalogue).

To manage the files of each dataset, there is the **Bookkeeping System**. Physicists can obtain from it a list of dataset file names that match their analysis requirements. There are data distribution policies to guarantee redundancy and availability according to demand, geographic distribution and security.

LFC metadata has a metadata file for each dataset name and its distribution around the world.

When EasyGrid submits a job using the clause *InputData* in the JDL file, only the CE with closest SE with data available will be selected. **VO tags** describing available software releases and packages complete the necessary information to distribute user's software to CEs

for processing.

The list of CEs defines the SEs that will store analysis software binary and large parametric files to minimize network traffic. EasyGrid performs all necessary procedures to replicate these files remotely and recover them efficiently.

The next stage is generation of all necessary information to submit the jobs on the Grid. **GEnerator of Resources Available** (GERA) produces the Job Description Language (JDL) files, the script with all necessary tasks to run the analysis remotely at a WN, and some grid dependent analysis parameters. The JDL files define the input sandbox with all necessary files to be transferred, and a WN balance load algorithm matches requirements to perform the task optimally.

When the **task is delivered in the WN**, scripts start running to initialize the specific environment, and user's software binary is downloaded from closest SE. Data files are made available through transfer or providing any access method to the application, and run user's software.

Users can **follow up** the process querying job status. If the job is done, a task recovering results in the user's directory is performed automatically. If the job was aborted in the process, the diagnostic listing is stored in the history file for further analysis.

Three benchmarks were developed. The first benchmark was eta(547) [8] reconstruction to test what is the best approach to data distribution: copy data file locally and read the file by application, or use a remote file access such as NFS. The software reads 1.4 gigabytes and produces several histograms for further analysis. Fig 1 shows the performance for different approaches in data access. In Figure 1a data is read directly from the local WN disk, in ideal conditions without overload and traffic. In Figure 1b it is first transferred from Storage Element. Transferring data produces an iowait in the initial part of the job due channel contention, and reading the data during execution looks better and more efficient.

However, this solution is not scalable as result of network's channel contention (Fig 1 c). Iowait increases to 50% and cpuload decreases to 50% with performance reduction.

The problem becomes even more significant when many nodes compete to access network (see Table I and II), increasing execution time from 600 s when data is local, to 2522 s with 12 cpus, and 6971 s with 56 cpus. The number of events analysed per second (EPS) decreases, which is a massive waste of resources, and shows the implemented paradigm may not be suitable for data grids because its efficiency is dependent on network availability.

The second benchmark was tau decays to neutral pions. This benchmark selected events over 482 million real events and generated 5 million MC events using EasyGrid [9].

The third benchmark was search for anti-deuterons in all events available in Run 3 (1,500 million events, in one week using 250 computers in parallel) [10].

There were no missing jobs, and few aborts were related with application problems. There were problems in grid catalogue when more then 250 jobs access at once.

3. Functional gridification algorithm.

Functional parallelism is a technique where functions in the program are executed independently in parallel more efficiently. There is a master program (or client) that request slave programs (or servers) for some service and coordinates effort and synchronization.

The **gridification algorithm** is a library with several functions to run conventional software on the grid doing functional parallelism, with minor changes in the source code. It is possible at same time apply data parallelism using EasyGrid.

The algorithm implements a master / slave architecture. The master manages a task queue that contains elementary tasks each slave can perform independently. One task can store data for a set of individual cases (service string) to overcome problems with communication delays between master/slaves. When one WN returns the results for a task, another task from the queue is send for processing.

The software was implemented using PVM commands [11], and can be changed to web services without any problem if necessary.

Functional gridification benchmark was genetic programming applied to evolve neutral pion discriminate function.

Genetic Programming (GP) [12]-[16] is an artificial intelligence algorithm that mimics the evolutionary process to obtain the best mathematical model given an economic function to evaluate performance (called the fitness function). Each fitness evaluation is an independent task to the gridification algoritm.

Genetic programming has been used to

determinate cuts to maximize event selection [17]-[19]. Genetic algorithms can also be associated with neural networks to implement discriminate functions [20] for Higgs boson.

Our approach is innovative because the mathematical model obtained with GP maps the variables hyperspace to a real value through the discrimination functions, an algebraic function of pions kinematics variables. Applying the discriminator to a given pair of gammas, if the discriminate value is bigger than zero, the pair of gammas is deemed to come from pion decay. Otherwise, the pair is deemed to come from another (background) source.

Two **datasets** were built, one for training with 57,992 records, and one for test with 302,374 records. Events with one real neutral pion were selected and marked as 1. Events without real pions and invariant mass reconstruction in the same region of real neutral pions where also selected and marked 0.

Kinematics data from each gamma used in the reconstruction were written in the datasets: angles of the gamma ray, 3-vector momentum, total momentum, and energy in the calorimeter. To avoid unit problems, we use sine, cosine and tangent values for each angle measured in the genetic trees. All other attributes are measured in GeV (1,000 million electron-volts).

Table III shows the results for training and test of 3 different runs. All results where in agreement and shows high purity, fundamental to study observable variables from neutral pion particles. High purity means there will be low non-neutral pions contamination in the sample (less then 10%). Efficiency of 83% means there will be a lost of 17% of real neutral pions from the sample, with decrease in number and increase of statistical error.

Table IV shows the time expended running standalone and with several numbers of slaves, with good performance: 10 slaves should reduce the time in ideal conditions to 10%, and our implementation achieved 24% despite all necessary communication overheads.

Conclusion.

In this paper implementations of data and functional parallelism using LCG/PVM grid environment are discussed and applied for several real case studies. A reliable job submission system (EasyGrid) manages all aspects of integration between user's requirements and resources for data grid. Functional gridification algorithm was implemented in client server architecture with good performance.

All software is available from the Internet [6], and is fully operational and easily adaptable for any application and experiment.

Discrimination functions can be used to discriminate neutral pions from background with 80% accuracy and 91% purity. This will allow me compare experimental values with observable obtained from theoretical Standard Model.

REFERENCE

[1] GridPP site: http://www.gridpp.ac.uk/

[2] The GridPP Collaboration: P J W Faulkner et al "GridPP: development of the UK computing Grid for particle physics" 2006 J. Phys. G: Nucl. Part. Phys. 32 N1-N20 doi:10.1088/0954-3899/32/1/N01

[3] CERN site: http://public.web.cern.ch/Public/Welcome.html

[4] LHC site: http://lhc.web.cern.ch/lhc/

[5] LCG site: http://lcg.web.cern.ch/LCG/

[6] Werner,J.C.; "HEP analysis, Grid and EasyGrid Job Submission Prototype: Babar/CM2 showcase" at http://www.hep.man.ac.uk/u/jamwer/

[7] BaBar Collaboration; "The BaBar detector", Nuclear Instruments and Methods in Physics Research A479(2002) 1-116.

[8] Tavera, M.; Private communication on eta reconstruction from TauUser data, PhD Thesis.

[9] Werner,J.C.; "Neutral Pion project" http://www.hep.man.ac.uk/u/jamwer/pi0alg5.html

[10] Werner,J.C.; "Search for anti-deuteron" http://www.hep.man.ac.uk/u/jamwer/deutdesc.html

[11] Parallel Virtual Machine site: http://www.csm.ornl.gov/pvm/pvm_home.html

[12] Holland,J.H. "Adaptation in natural and artificial systems: na introductory analysis with applications to biology, control and artificial intelligence." Cambridge: Cambridge press 1992.

[13] Goldberg,D.E. "Genetic Algorithms in Search, Optimisation, and Machine Learning." Reading, Mass.: Addison-Whesley, 1989.

[14] Chambers,L.; "The practical handbook of Genetic Algorithms" Chapman & Hall/CRC,2000.

[15] Koza,J.R. "Genetic programming: On the programming of computers by means of natural selection." Cambridge,Mass.: MIT Press, 1992.

[16] Werner,J.C.; "Active noise control in ducts using genetic algorithm" PhD. Thesis- São Paulo University-São Paulo-Brazil-1999.

[17] Cranmer,K.; Bowman,R.S.; "PhysicsGP: A genetic programming approach to event selection" Computer Physics Communications 167 (2005) 165-176.

[18] Focus Collaboration, "Application of genetic programming to high energy physics event selection" Nuclear instruments and methods in physics research A 551 (2005) 504-527.

[19] Focus Collaboration; "Search for L+c -> pK+p- and D+s -> K+K+p- using genetic programming event selection" Physics letters B 624 (2005) 166-172

[20] Mjahed, M.; "Search for Higgs boson at LHC by using genetic algorithms" To be published in Nuclear Instruments and Methods in Physics Research.

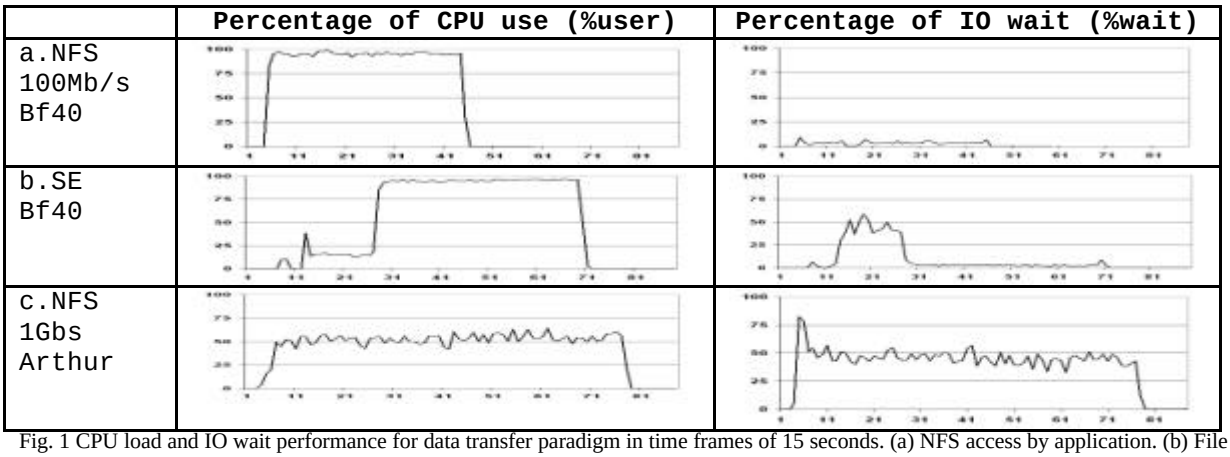


Fig. 1 CPU load and IO wait performance for data transfer paradigm in time frames of 15 seconds. (a) NFS access by application. (b) File copy locally and later access. (c) NFS access with jammed network.

TABLE I DATA DISTRIBUTION ANALYSIS. PERFORMANCE WAS DEFINED AS (100 * EPS/EPS_LOCAL), WHERE EPS IS EVENTS PER SECOND. EPS_LOCAL (1577) IS NUMBER OF EVENTS PER SECOND, USING LOCAL STORED FILES AND TAKES 600 SECONDS.

# Jobs	c.NFS 1 Gbps		a.NFS		b.SE 100Mbps	
	EPS	Perf	EPS	Perf	EPS	Perf
1	855	54	1574	100	1193	76
3	481	31	1457	92	949	60
6	492	31	1388	88	725	46
12	412	26	1372	87	495	31

TABLE II: AVERAGE EXECUTION TIME OF 500 JOBS IN 12 AND 56 CPUS IN PARALLEL WITH FILE TRANSFER AND REMOTE ACCESS.

500 Jobs		File Transfer	Exec	Total
12 CPUs	Time	00:15:00	00:27:00	00:42:02
	Seconds	900	1620	2522
56 CPUs	Time	01:43:52	00:12:19	01:56:11
	Seconds	6232	739	6971

TABLE III TRAINING AND TESTS RESULTS FROM DISCRIMINATE FUNCTION OBTAINED USING GENETIC PROGRAMMING WITH DIFFERENT DATASETS.

		Case 1:Forecast		Case 2: Forecast		Case 3: Forecast	
Training 57992 records	Real	D>0	D<0	D>0	D<0	D>0	D<0
	1	23299	4819	23368	4750	22491	5627
	0	3093	26781	3040	26834	2731	27143
	Accuracy	86		86		85	
	Efficiency	82		83		80	
	Purity	89		89		90	
Test 302374 records	Real	D>0	D<0	D>0	D<0	D>0	D<0
	1	117268	41037	117215	41090	111999	46306
	0	14153	129916	13870	130199	12543	131526
	Accuracy	81		81		80	
	Efficiency	74		74		70	
	Purity	90		90		91	

TABLE IV EXECUTION TIME FOR THE SAME SOFTWARE WITH DIFFERENT NUMBER OF SLAVES AND NODES.

	Standalone	1node / 2 slaves	5 nodes / 10 slaves
Time(1,000s)	80	47	19
Improvement		58%	24%

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Building a distributed software environment at UCL utilising a switched light path

V. Bartsch[1], N. Pezzi[1], M. Lancaster[1]
[1]University College London

July 5, 2006

Abstract

The amount of data produced in high energy experiments like CDF (Collider Detector at Fermilab) [1] requires to distribute the computing and the data for the scientists analysing data of the experiments. Often remote computing resources are shared throughout several high energy experiments, although the underlying grid software differs from experiment to experiment. Here the situation is described for the computing set up at UCL (University College London) for both the local users and all users for the experiment CDF. Grid software and techniques developed for the CDF experiment are used and a dedicated switched light path between Fermilab and UCL is utilised.

1 Introduction

CDF is a particle physics experiment investigating the fundamental nature of matter. It is presently taking data from proton anti-proton collisions at the Tevatron at Fermilab, which is located just outside Chicago in the USA. The experiment currently produces approximately 1 PB of raw data per year and will continue to do so until 2009. Analysis of this data is underway by almost 800 physicists located at 61 institutions in 13 countries across 3 continents. The amount of raw data and the need to produce secondary reduced datasets have required new distributed storage and analysis. Grid systems based on DCAF [2] and SAM [4] have been developed and deployed during the last year and the focus is now to make the DCAF systems interoperable with other grid systems, e.g. LCG [5] sites. The aim is that 50% of CDF's CPU and storage requirements will be provided by institutions remote from Fermilab. In order to effectively utilise this distributed computing network it is necessary to have high speed point to point connections, particularly to and from Fermilab, which have a bandwidth significantly higher than commonly available. To this end, as part of the ESLEA [6] project, the use of a dedicated switch light path from Fermilab to UCL in the UK has been optimised.

This paper describes the experiences utilising grid middleware in a switched lightpath environment for the CDF experiment.

2 CDF's Data Handling and Analysis Systems

The analysis model of data in high energy physics is highly sequential and is generally carried out on dedicated Linux PC analysis farms. These farms may be shared between high energy physics experiments. Since there are still several approaches to implement a world wide computing grid it becomes more and more important to design grid systems in a way that they are interoperable with each other. The CCC grid cluster at UCL [7] is a CERN Tier2 center which utilises the LCG approach to the grid. CDF however chose to use a combination of the SAM and the DCAF system, which are explained subsequently, as a grid system.

The core capabilities of the grid enabled data handling system SAM are bookkeeping

of metadata and locations of data files and transfers of the files to the nodes to process them. In order to be used at a remote site several services need to be running remotely which are summarised as a SAM station. The SAM station communicates with a central service, the so-called database server via CORBA in order to get the location of a file and is able to transfer the file with various transfer mechanisms including gridftp. The new location is reported back to the central database. Fig. 1 shows the data consumption of all SAM stations last year. The major amount of data has been consumed at FNAL because SAM is used both onsite and remotely. The SAM stations at UCL have started to import data beginning of this year due to a downtime of the UK light link autumn until winter last year.

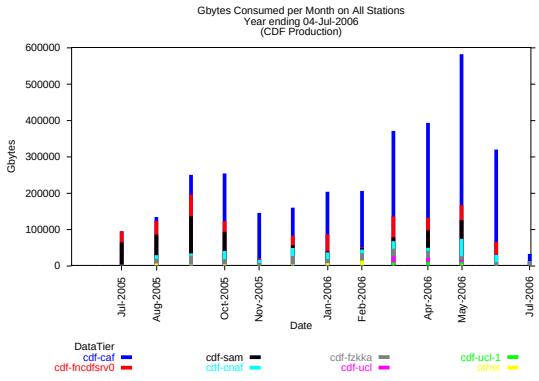


Figure 1: *Data consumption of all SAM stations last year. cdf-caf, cdf-fncdfsrv0 and cdf-sam are local to Fermilab, cdf-cnaf is located at Bologna (Italy), cdf-fzkka at Karlsruhe (Germany), cdf-ucl and cdf-ucl-1 at London (United Kingdom).*

DCAF systems provide authorization mechanisms (currently kerberos) and job handling mechanism to run CDF jobs in case the CDF software is NFS mounted on all worker nodes. In order to run on LCG clusters the job is submitted to the globus gatekeeper at the LCG headnode and uses during runtime the Condor [8] glidein mechanism by which one or more grid resources temporarily join a local Condor pool. The so-called GlideCaf is described in more detail at [3]. The features of this pool

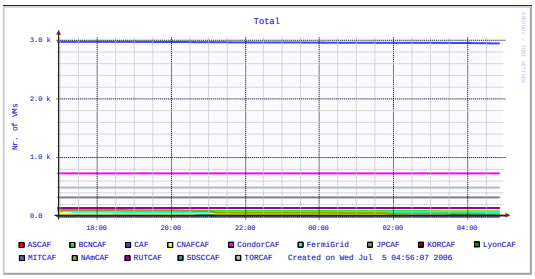


Figure 2: *Total number of virtual machines (VMs) at the DCAFs during the last year: The main one are CAF, CondorCAF, FermiGrid at FNAL. JPCAF, KorCAF and ASCAF are located in Asia, NAmCAF, TORCAF and MITCAF in the US/Canada, CNAFCAF in Italy.*

are changed so that it appears like any DCAF cluster. In order to utilise a 200 PC cluster at UCL this technique will be used, it has already been tested at a smaller cluster at UCL.

Fig. 2 shows the total number of virtual machines (VMS) at each CAF worldwide. The main resources (about 4kVMs) can still be found at FNAL, about 2k VMs are located outside of FNAL. Some of the DCAFs are not using the LCG submission mechanism but dedicated resources (e.g. ASCAF and JPCAF), others use the glidein mechanism, for example LyonCAF and FermiGrid. One can see that those resources claim not to have any CPUs at all, but when a job gets submitted they are executing through the Globus gatekeeper and therefore running on a cluster which does not appear in the monitoring. Fig. 3 shows the usage of the DCAFs throughout the last year. One can see that the central resources show a much higher usage than the remote DCAFs. This is partly due to the fact that the import of necessary data from FNAL takes long, partly due to downtimes of the resources. The UCL glidein CAF is not yet officially monitored.

3 Utilisation of the UK/Star Light switched light link

Typical CDF secondary datasets are presently 1-50 TB in size. The CPU resources required to repeatedly analyse such datasets will ex-

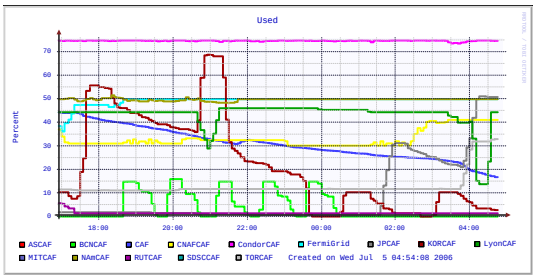


Figure 3: *Usage of the DCAFs in percentage during the last year.*

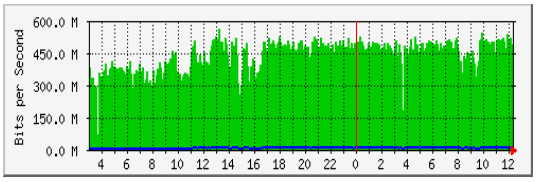


Figure 4: *Snapshot of the data transfer rate at the UK light link during one day. A sustained rate of 550 Mbit/sec is feasible.*

ceed those available at Fermilab and so the datasets need to be distributed to centers in Europe and Asia to facilitate the use of their CPU resources. Typical transfer rates from Fermilab to Europe (UCL) using the standard network are approximately 25 Mbit/sec (for multiple streams). The transfer of a single dataset would take months. Therefore most DCAFs are specializing in dedicated branches of physics and the according datasets. UCL has the possibility to transfer data over the dedicated switched light path from Fermilab (using the US Starlight network [9]) to UCL in the UK (using the UKLight network [10]) with a bandwidth of 1 Gbit/sec. During the last year several integrity and bandwidth tests have been performed which showed that network equipment in the path between the CDF storage area at Fermilab to the Starlight network was causing an unexpected slowdown and therefore these network switches needed to be replaced. After this replacement sustained data transfer rates from the CDF storage at Fermilab to the disks at UCL have been at a maximum of 550 Mbit/sec at a 2 hours average. Typical data transfer rates are shown at Fig. 4. Therefore it is possible for the users to import datasets on the fly rather than the site administrator caring for the transfer.

4 Conclusion

The software setup allows users from UCL and CDF to utilize the computing environment of the local cluster at UCL. A further extension of the services of the Tier-2 center at UCL in on the way. The dedicated switched light link between Fermilab and the UK delivers a sustained data transfer rate from disk to disk of 550 Mbit/sec and therefore allows the users to choose the data to analyse without high delay due to data transfer. This is unique compared to other remote data analysis centers of CDF which allow to run only on the data which was copied due to the decision of the site administrators.

References

[1] http://www-cdf.fnal.gov/

[2] http://cdfcaf.fnal.gov

[3] GlideCaf - a late binding approach to the Grid, I. Sfiligoi et al., CHEP06 conference proceedings

[4] http://d0db-prd.fnal.gov/sam/

[5] http://lcg.web.cern.ch/LCG/

[6] http://www.eslea.uklight.ac.uk

[7] http://wiki.gridpp.ac.uk/wiki/UCL-CCC

[8] http://www.cs.wisc.edu/condor/

[9] http://www.startap.net/starlight/

[10] http://www.uklight.ac.uk

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Collaborative Visualization of 3D Point Based Anatomical Model within a Grid Enabled Environment

Ian Grimstead, David Walker and Nick Avis; School of Computer Science, Cardiff University
Frederic Kleinermann; Department of Computer Science, Vrije Universiteit
John McClure; Directorate of Laboratory Medicine, Manchester Royal Infirmary

Abstract

We present a use of Grid technologies to deliver educational content and support collaborative learning. "Potted" pathology specimens produce many problems with conservation and appropriate access. The planned increase in medical students further exasperates these issues, prompting the use of new computer-based teaching methods. A solution to obtaining accurate organ models is to use point-based laser scanning techniques, exemplified by the Arius3D colour scanner. Such large datasets create visually compelling and accurate (topology and colour) virtual organs, but will often overwhelm local resources, requiring the use of specialised facilities for display and interaction. The Resource-Aware Visualization Environment (RAVE) has been used to collaboratively display these models, supporting multiple simultaneous users on a range of resources. A variety of deployment methods are presented, from web page applets to a direct AccessGrid video feed.

1. Introduction

Medical education has to adapt and adjust to recent legislation and the reduction in the working hours of medical students. There is already some evidence that these changes are impacting negatively on students' anatomical understanding [Ellis02, Older04].

1.1. Pathology Specimens

The University of Manchester Medical School has over 5,000 "potted" pathology specimens (3,000 on display at any one time) at three sites, which is typical of a large UK teaching school. These specimens have been collected, preserved and documented over a number of years, representing a valuable "gold standard" teaching resource. The restrictions on harvesting and use of human tissue, coupled with the problems and logistics associated with the conservation of the collections, result in large issues for the planned increase in medical students; something has to change.

New computer based teaching methods and materials are appearing which include anatomical atlases to assist the students' understanding of anatomy, moving away from the more traditional teaching materials. The availability and use of the Visible Human dataset is a good example of how computer based teaching is predicated on the availability of high quality primary data. However, the construction of high quality teaching material such data requires tremendous amounts of expert human intervention. There is therefore a compelling need to discover ways of simply and quickly creating and sharing digital 3D "virtual organs" that can be used to aid the teaching of anatomy and human pathology. This is the motivation of our studies.

1.2. Point-Based Models

It is difficult and time consuming to create a library of 3D virtual organs to expose the trainee to a range of virtual organs that represent biological variability and disease pathology. Furthermore, such virtual organs must be both topologically correct and also exhibit colour consistency to support the identification of pathological lesions.

A solution to obtaining accurate organ models is to use a point-based laser scanner. The Arius3D colour laser system is unique, as far as we are aware, in that it recovers both topology and colour from the scanned object. The Arius3D solution is also compelling as it produces an accurate and visually compelling 3D representation of the organ without resorting to more traditional texture mapping techniques.

Whilst the above method eases the capture and generation of accurate models, the resulting file sizes and rendering load can quickly overwhelm local resources. Furthermore a technique is also required to distribute the models for collaborative investigation and teaching, which can cope with different types of host platform without the user needing to configure the local machine.

2. RAVE – the Resource-Aware Visualization Environment

The continued increases in network speed and connectivity are promoting the use of remote resources via Grid computing, based on the

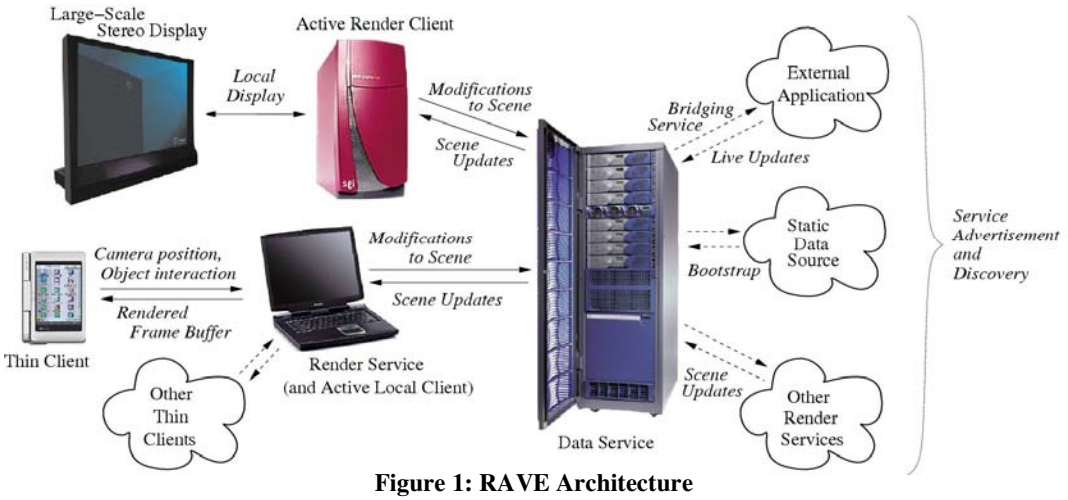


Figure 1: RAVE Architecture

concept that compute power should be as simple to access as electricity on an electrical power grid – hence "Grid Computing". A popular approach for accessing such resources is Web Services, where remote resources can be accessed via a web server hosting the appropriate infrastructure. The underlying machine is abstracted away, permitting users to remotely access different resources without considering their underlying architecture, operating system, etc. This both simplifies access for users and promotes the sharing of specialised equipment.

The RAVE (Resource-Aware Visualization Environment) is a distributed, collaborative visualization environment that uses Grid technology to support automated resource discovery across heterogeneous machines. RAVE runs as a background process using Web Services, enabling us to share resources with other users rather than commandeering an entire machine. RAVE supports a wide range of machines, from hand-held PDAs to high-end servers with large-scale stereo, tracked displays. The local display device may render all, some or none of the data set remotely, depending on its capability and present loading. The architecture of RAVE has been published elsewhere [Grims04], so we shall only present a brief overview of the system here.

2.1. Architecture

The Data Service is the central part of RAVE (see Figure 2), forming a persistent and centralised distribution point for the data to be visualized. Data are imported from a static file or a live feed from an external program, either of which may be local or remotely hosted. Multiple sessions may be managed by the same Data Service, sharing resources between users. The data are stored in the form of a scene tree; nodes of the tree may contain various types of data, such as voxels, point clouds or polygons. This enables us to support different data formats for visualization, which is particularly important for the medical applications reported here.

The end user connects to the Data Service through an Active Client, which is a machine that has a graphics processor and is capable of rendering the dataset. The Active Client downloads a copy of (or a subset of) the latest data, then receiving any changes made to the data by remote users whilst sending any local changes made by the local user, who interacts with the locally rendered dataset.

If a local client does not have sufficient resources to render the data, a Render Service can be used instead to perform the rendering and send the rendered frame over the network to the client for display. Multiple render sessions are supported by each Render Service, so multiple users may share available rendering resources. If multiple users view the same session, then single copies of the data are stored in the Render Service to save resources.

2.2. Point-Based Anatomical Models

To display point-based models in RAVE, the physical model is first converted into a point cloud dataset via an Arius3D colour laser scanner, retaining surface colour and normal information. This is stored as a proprietary PointStream model, before conversion to an open format for RAVE to use. The three stages are shown in Figure 3, presenting a plastinated human heart; point samples were taken every 200μm producing a model of approximately 6 million datapoints.

The flexible nature of RAVE enables this large dataset to be viewed collaboratively by multiple, simultaneous users – irrespective of their local display system. For instance, a PDA can view and interact with the dataset alongside a high-end graphical workstation.

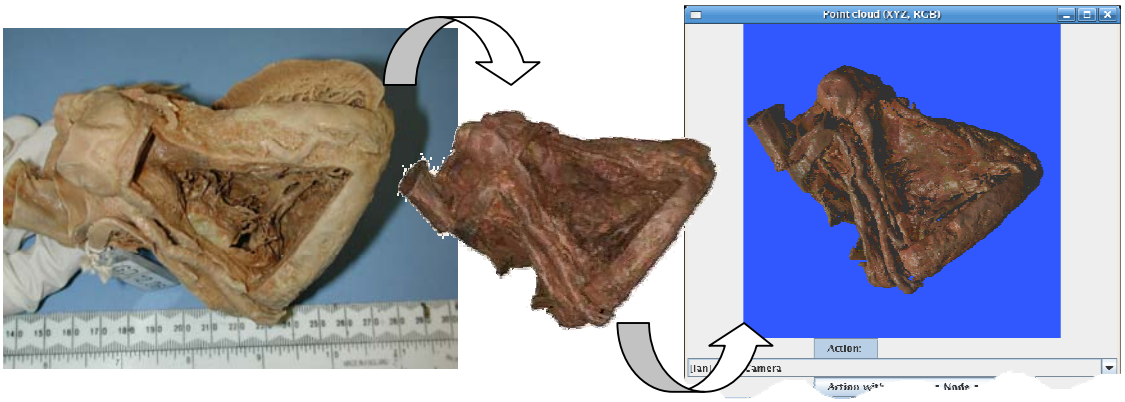


Figure 4: Plastinated human heart, data viewed from PointStream and corresponding RAVE client

2.3. Collaborative Environment

RAVE supports a shared scenegraph; this basically means that the structure representing the dataset being viewed is actually shared between all participating clients. Hence if one client alters the dataset, the change is reflected to all other participating clients. A side-effect of this is that when a client navigates around the dataset, other users can see the client's "avatar" also move around the dataset (an avatar is an object representing the client in the dataset). Clients can then determine where other clients are looking/located and can opt to fly to other client's positions.

If a client discovers a useful artefact in a dataset, they can leave 3D "markers" for other users, together with a textual description entered by the client. These can be laid in advance by a client (such as a teacher or lecturer), leaving a guided trail for others to follow.

RAVE also supports AccessGrid (AG) [Childers00], which was demonstrated at SAND'05 [Grims05]. A RAVE thin client was running in Swansea, along with an AG client. An AG feed was also launched from a Render Service in Cardiff which was then controlled from the Thin Client in Swansea.

2.4. Ease of Use

To simplify the use of RAVE, we have created a Wizard-style Java applet that can be launched stand-alone or via a web browser. The applet uses UDDI to discover all advertised RAVE data services; these are then interrogated in turn to obtain all hosted data sessions (and hence datasets). The GUI also enables the user to launch a client, and contains a notification area to see progress messages.

To collaboratively view a dataset, the user simply selects the dataset in the GUI, enters the size of the visual window they require, the name associated with their avatar and the minimum frames-per-second they are willing to accept. The Wizard then determines if the local host can display the data as an Active Client; if so, it verifies there is sufficient memory to host the data and that local resources can produce the requested frames per second..

If insufficient resources exist at the local host, then a Thin Client will be used instead. The applet searches (via UDDI) for available Render Services; for each responding Render Service, we examine the available network bandwidth to the client, memory available for datasets, remote render speed and if the requested dataset is already hosted on this service. The most appropriate Render Service is then selected, and a Thin Client is spawned in a new window.

Render Services contain resource-awareness for Thin Client support. During image render and transmission at the Render Service, the image rendering time is compared with the image transmission time. Image compression over the network is adjusted to match transmission time with render time, to maintain the maximum possible framerate. As this will induce lossy compression, an incremental codec is used; if the image does not change, then the lossy nature is reduced, building on previous frames to produce a perfect image over subsequent frame transmissions (forming a golden thread as [Bergman86]).

3. Discussion and Conclusion

We have presented systems and methods to allow the creation of 3D virtual organs and their use in a collaborative, distributed teaching environment through the use of various Grid technologies.

The availability of high quality datasets is the beginning of the process in creating compelling and useful anatomical teaching

tools. Whilst commodity graphics cards are increasing in their performance at more than Moore's Law rates, their ability to handle large complex datasets still presents barriers to the widespread use of this material. This coupled with slow communications links between centralised resources and students who are often working remotely and in groups (using problem based teaching methods) and these facts quickly erode some of the advantages of digital assets over their physical counterparts.

To this end we have sought to harness grid middleware to remove the need for expensive and high capability local resources – instead harnessing remote resources in a seamless and transparent manner to the end user to effect a compelling and responsive learning infrastructure.

We have presented methods to allow both the quick and relatively easy creation of 3D virtual organs and their visualization. We maintain that taken together these developments have the potential to change the present anatomical teaching methods and to hopefully promote greater understanding of human anatomy and pathology.

4. References

[Avis00] N. J. Avis. *Virtual Environment Technologies*, Journal of Minimally Invasive Therapy and Allied Technologies, Vol 9(5) pp333- 340, 2000.

[Avis04] Nick J Avis, Frederic Kleinermann and John McClure. *Soft Tissue Surface-Scanning: A Comparison of Commercial 3D Object Scanners for Surgical Simulation Content Creation and Medical Education Applications*, Medical Simulation, International Symposium, ISMS 2004, Cambridge, MA, USA, Springer Lecture Notes in Computer Science (3078), Stephane Cotin and Dimitris Metaxas (Eds.), pp 210-220, 2004.

[Bergman86] L. D. Bergman, H. Fuchs, E. Grant, and S. Spach. *Image rendering by adaptive refinement*. Computer Graphics (Proceedings of SIGGRAPH 86), 20(4):29--37, August 1986. Held in Dallas, Texas.

[Childers00] Lisa Childers, Terry Disz, Robert Olson, Michael E. Papka, Rick Stevens and Tushar Udeshi. *Access Grid: Immersive Group-to-Group Collaborative Visualization*, in Proceedings of the 4th International Immersive Projection Technology Workshop, Ames, Iowa, USA, 2000.

[Ellis02] H. Ellis. *Medico-legal litigation and its links with surgical anatomy*. Surgery 2002.

[Grims04] Ian J. Grimstead, Nick J. Avis, and David W. Walker. *Automatic Distribution of Rendering Workloads in a Grid Enabled Collaborative Visualization Environment*, in Proceedings of Supercomputing 2004, held 6th-12th November in Pittsburgh, USA, 2004.

[Grims05] Ian J. Grimstead. *RAVE – The Resource-Aware Visualization Environment*, presentation at SAND'05, Swansea Animation Days 2005, Taliesin Arts Centre, Swansea, Wales, UK, November 2005.

[Nava03] A. Nava, E. Mazza, F. Kleinermann, N. J. Avis and J. McClure. *Determination of the mechanical properties of soft human tissues through aspiration experiments*. In MICCAI 2003, R E Ellis and T M Peters (Eds.) LNCS (2878), pp 222-229, Springer-Verlag, 2003.

[Older04] J. Older. *Anatomy: A must for teaching the next generation*, Surg J R Coll Edinb Irel., 2 April 2004, pp79-90

[Vidal06] F. P. Vidal, F. Bello, K. W. Brodlie, D. A. Gould, N. W. John, R. Phillips and N. J. Avis. *Principles and Applications of Computer Graphics in Medicine*, Computer Graphics Forum, Vol 25, Issue 1, 2006, pp 113-137.

5. Acknowledgements

This study was in part supported by a grant from the Pathological Society of Great Britain and Ireland and North Western Deanery for Postgraduate Medicine and Dentistry. All specific permissions were granted regarding the use of human material for this study.

We also acknowledge the support of Kestrel3D Ltd and the UK's DTI for funding the RAVE project. We would also like to thank Chris Cornish of Inition Ltd. for his support with dataset conversion.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Alternative Security Architectures for e-Science*

Jason Crampton, **Hoon Wei Lim**, Kenneth G. Paterson, and Geraint Price
Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK

Abstract

There have been recent proposals in search of alternative security architectures to PKIs for grid application and e-Science. The application of identity-based cryptography (IBC) in designing a grid security architecture seems to be interesting because of its attractive properties, such as, being certificate-free and having small key sizes. In this paper, we discuss our latest research findings of the identity-based approach in designing a grid security architecture. These include a performance analysis of the identity-based approach, and the application of identity-based secret public keys in designing a password-based version of the standard TLS protocol used in MyProxy.

1 Introduction

The majority of current grid security implementations are based on public key infrastructure (PKI) [6, 15], and the operational grid developed as part of the UK e-Science project is no exception. However, large-scale PKIs are known to have many problems which have hindered the widespread adoption of PKI technology [8, 11]. These include cost, scalability (both of registration and key management processes), revocation, management of client private keys, and support for dynamic security policies.

In addition to generic PKI problems, grid-specific security requirements bring further concerns. The Globus Toolkit (GT) [5], the *de facto* standard for building grids, includes the specification of a security architecture, the Grid Security Infrastructure (GSI) [6]. This in turn makes use of proxy certificates [16], in addition to the standard X.509 public key certificates, to support single sign-on and delegation services. The dependence on proxy certificates causes many performance issues. Their use leads to frequent and computationally expensive RSA key-pair generation, and the consumption of bandwidth and processing power for transmitting and checking the lengthy certificate chains that result. Moreover, the delegation protocol that is used involves a round-trip between a delegator and a delegation target, and consequent delay. The set of cryptographic algorithms that can be used is quite limited, with RSA signatures being dominant. These issues may not be serious problems for today's grid environments, but they may limit the spread of grid technology to pervasive and mobile environments, where devices lack computational power and the communication networks have limited bandwidth.

Independent of grid computing, a variant of traditional public key technologies called identity-based cryptography (IBC) [4, 13] has recently received considerable attention. Through IBC, an identifier which represents a user can be transformed into his public key and used on-the-fly without any certificate checking. The potential of IBC to provide greater flexibility to entities within a security infrastructure and its certificate-free approach may well match the dynamic qualities of grid environments. We proposed a fully identity-based key infrastructure for grid (IKIG) [9] which meets the security requirements of the GSI. The proposal makes use of both long-term and short-term identity-based keys, by exploiting some properties from hierarchical identity-based cryptography (HIBC) [7]. More details about IKIG will be presented in Section 2.

In this paper, we intend to demonstrate our latest results of some follow-on work from [9]. These include: (i) performance analysis based on actual implementations of the cryptographic schemes adopted in [9], and (ii) the application of identity-based secret public keys in designing a password-based TLS protocol, which in turn seems to be suited to the authentication protocol used by MyProxy [3].

2 Identity-Based Key Infrastructure for Grid

2.1 Overview

One motivation for the proposal of an identity-based key infrastructure for grid (IKIG) of [9] is the attractive properties of IBC. These include:

*This research was supported by the EPSRC under grant EP/D051878/1.

- *Identity-based*: The use of identity-based public keys in IBC allows any entity's public key to be generated and used on-the-fly;
- *Certificate-free*: IBC does not make use of certificates since public keys can be computed based on some public identifiers; and
- *Small key sizes*: Since identity-based cryptographic schemes use pairings which are, in turn, based on elliptic curves, they can have smaller key sizes than more conventional public key cryptosystems such as RSA.

By exploiting some properties from HIBC, IKIG facilitates the creation and usage of identity-based proxy credentials in a very natural way. These identity-based proxy credentials, in turn, are needed to support features that match those provided by the GSI.

In the IKIG setting, the roles of a Certificate Authority (CA) in the current PKI-based GSI has been replaced by a Trusted Authority (TA). The TA's roles including acting as the Private Key Generator (PKG) and supporting other user-related administration. Figure 1 shows the hierarchical setting of HIBC that matches the hierarchical relationships of various entities within a grid environment, with the TA at level 0, user at level 1 and user proxy at level 2.

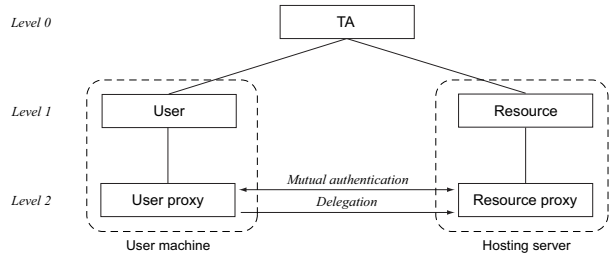


Figure 1. A hierarchical structure of entities in the IKIG setting.

We note that proxy credentials must be used for secure job submissions in order to match the requirements of the GSI. These short-term credentials are used in security services such as mutual authentication and delegation. The TA distributes long-term private keys to its users (and resource providers) at level 1, who in turn generate short-term private keys for their own proxies at level 2, as illustrated in Figure 1.

IKIG has the following features:

1. Single sign-on: As with the GSI, our IKIG proposal supports single sign-on through the use of identity-based proxy credentials. Since a user's short-term public keys are based on some predictable identifiers and the matching short-term private keys are stored locally at the user side, user authentication can be performed without any physical intervention from the users and without the need for certificates.
2. Mutual authentication and key agreement: IKIG also supports a certificate-free authenticated key agreement protocol based on the TLS handshake. Our protocol allows mutual authentication and session key establishment between two entities in a more lightweight manner than the traditional TLS as it has small key sizes and requires no certificates.
3. Delegation: We propose a non-interactive delegation protocol which works in the same way as in the GSI, in the sense that the delegator signs a new public key of the delegation target. In addition, IKIG allows partial aggregation of signatures. This can be useful when the length of the delegation chain increases. This is a new feature not available in the GSI.

Users in the IKIG setting do not need to obtain short-term private keys from their respective PKGs. This is because the users themselves act as PKGs for their local proxy clients. Thus short-term private key distribution is not an issue in IKIG. This contrasts favourably with conventional applications of IBC, where private key distribution is a complicating factor.

2.2 Performance Analysis

We examined the efficiency of IKIG in terms of its communication and computational overheads.

Since identity-based cryptographic schemes have small key sizes and does not rely on the use of certificates, it is obvious that IKIG consumes much less bandwidth than the RSA-based GSI does. More details of the performance trade-offs in communication costs between the GSI and IKIG can be found in [9].

To obtain more accurate real performance figures in terms of computation times of the underlying cryptographic schemes of IKIG, we have recently implemented the Gentry-Silverberg hierarchical encryption and signature schemes (HIBE/HIBS). Our implementation was based on the MIRACL library [14], written in C/C++ and compiled with Microsoft Visual C++ 6.0. By using known optimisation techniques, we observed that the computational cost of IKIG is comparable to the GSI, even though pairing computations are generally perceived to be computationally intensive. The details of the computation timings of the cryptographic schemes used in the GSI and IKIG are shown in Table 1.

It is worth noting that recent results (see for example [2, 12]) have shown improvements in computing pairings with the use of various optimisation techniques and

Table 1. Performance trade-offs in computation times (in milliseconds) between the GSI and the IKIG settings on a Pentium IV 2.4 GHz machine.

	GSI		IKIG	
Operation	**RSA**	**Time**	**HIBE/HIBS**	**Time**
Key generation				
(a.) Long-term	1 GEN	149.90	1 EXT	1.69
(b.) Short-term	1 GEN	34.85	1 EXT	1.74
Authenticated key agreement				
(a.) Requestor	1 1024-bit VER 1 512-bit ENC 1 512-bit SIG 1 512-bit VER	2.67	1 ENC, 1 SIG	8.79
(b.) Resource	1 1024-bit VER 1 512-bit DEC 2 512-bit VERs	2.67	1 DEC, 1 VER	20.16
Delegation				
(a.) Delegator	1 512-bit SIG 1 512-bit VER	1.86	1 SIG	3.35
(b.) Delegation target	1 GEN 1 512-bit SIG	35.63	1 EXT	1.74
(c.) Verifier	3 512-bit VERs	0.84	1 VER	8.42

GEN = RSA parameter generation EXT = HIBE/HIBS private key extraction
ENC = Encryption DEC = Decryption
SIG = Signing VER = Verification

this should give hope to faster HIBE and HIBS schemes in the near future. On the other hand, we remark that it is unlikely that significant algorithmic improvements for RSA computations will be forthcoming, since this field has been intensively researched for many years.

2.3 Identity-Based Secret Public Keys

In a password-based authentication protocol, a secret public key is a standard public key which can be generated by a user or an authentication server, and is known only to them but is kept secret from third parties. A secret public key, when encrypted with a user's password, should serve as an unverifiable text. This may significantly increase the difficulty of password guessing even if it is a poorly chosen password as an attacker has no way to verify if he has made the correct guess. However, it may not be easy to achieve unverifiability of text by simply performing naive symmetric encryption on public keys of standard types, such as RSA or ElGamal, which contain certain number theoretic structure.

We have recently investigated the use of identity-based secret public keys [10] and designed a password-based TLS protocol which, in turn, seems to fit nicely into MyProxy. The identity-based approach of [10] shows that secret public keys can be constructed in a very natural way using arbitrary random strings, eliminating the structure found in, for example, RSA or ElGamal keys.

An identity-based secret public key protocol can be naturally converted into a password-based version of the standard TLS protocol. The resulting protocol allows passwords to be tied directly to the establishment of secure TLS channels (rather than transmitting plain passwords through secure TLS channels). Such a protocol can be constructed without radical modification to the existing TLS handshake messages. The identity/password-based TLS protocol is directly applicable to securing interactions between grid users and MyProxy in IKIG.

3 Future Work

Certificateless public key cryptography (CL-PKC) [1] offers an interesting combination of features from IBC and standard public key cryptography. In particular, each user selects a public/private key-pair (in addition to a TA-generated private key component that matches an identifier), thereby eliminating the problem of key escrow found

in IBC. Moreover, the public keys do not need to be supported by certificates, so as with IBC, many of the problems associated with certificate management in PKI are eliminated in CL-PKC.

As part of our future work, we plan to study the application of CL-PKC in grid environments. We will develop a security architecture for grid systems based on CL-PKC that parallels our existing work on IKIG. We will examine how CL-PKC can be used to support authentication and establishment of secure communications via a TLS-like protocol. We will also study the key management aspects of our architecture, such as, key updating, key revocation and the use of credential storage systems, such as MyProxy, in our architecture.

Subsequently, we plan to conduct a performance analysis to compare our CL-PKC approach with the existing GSI and IKIG approaches.

References

[1] S.S. Al-Riyami and K.G. Paterson. Certificateless public key cryptography. In C.S. Laih, editor, *Advances in Cryptology - Proceedings of ASIACRYPT 2003*, pages 452–473. Springer-Verlag LNCS 2894, 2003.

[2] P. S. L. M. Barreto, S. D. Galbraith, C. Ó hÉigeartaigh, and M. Scott. *Efficient Pairing Computation on Supersingular Abelian Varieties*. Cryptology ePrint Archive, Report 2004/375, September 2005. Available at http://eprint.iacr.org/2004/375.

[3] J. Basney, M. Humphrey, and V. Welch. The MyProxy online credential repository. *Journal of Software: Practice and Experience*, 35(9):817–826, July 2005.

[4] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *Advances in Cryptology - Proceedings of CRYPTO 2001*, pages 213–229. Springer-Verlag LNCS 2139, 2001.

[5] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputing Applications*, 11(2):115–128, 1997.

[6] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational Grids. In *Proceedings of the 5th ACM Computer and Communications Security Conference*, pages 83–92. ACM Press, 1998.

[7] C. Gentry and A. Silverberg. Hierarchical ID-Based cryptography. In Y. Zheng, editor, *Advances in Cryptology - Proceedings of ASIACRYPT 2002*, pages 548–566. Springer-Verlag LNCS 2501, 2002.

[8] P. Gutmann. PKI: It's not dead, just resting. *IEEE Computer*, 35(8):41–49, August 2002.

[9] H.W. Lim and K.G. Paterson. Identity-based cryptography for grid security. In H. Stockinger, R. Buyya, and R. Perrott, editors, *Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing (e-Science 2005)*, pages 395–404. IEEE Computer Society Press, 2005.

[10] H.W. Lim and K.G. Paterson. Secret public key protocols revisited. In *Proceedings of the 14th International Workshop on Security Protocols 2006*, to appear.

[11] G. Price. PKI challenges: An industry analysis. In J. Zhou, M-C. Kang, F. Bao, and H-H. Pang, editors, *Proceedings of the 4th International Workshop for Applied PKI (IWAP 2005)*, pages 3–16. Volume 128 of FAIA, IOS Press, 2005.

[12] M. Scott. Computing the Tate pairing. In A. Menezes, editor, *Proceedings of the RSA Conference: Topics in Cryptology - the Cryptographers' Track (CT-RSA 2005)*, pages 293–304. Springer-Verlag LNCS 3376, 2005.

[13] A. Shamir. Identity-based cryptosystems and signature schemes. In G.R. Blakley and D. Chaum, editors, *Advances in Cryptology - Proceedings of CRYPTO'84*, pages 47–53. Springer-Verlag LNCS 196, 1985.

[14] Shamus Software Ltd. *MIRACL*. Available at http://indigo.ie/∼mscott/, last accessed in April 2006.

[15] M.R. Thompson and K.R. Jackson. Security issues of grid resource management. In J. Weglarz, J. Nabrzyski, J. Schopf, and M. Stroinski, editors, *Chapter 5 of Grid Resource Management: State of the Art and Future Trends*, pages 53–69, Boston, 2003. Kluwer Academic.

[16] S. Tuecke, V. Welch, D. Engert, L. Pearman, and M. Thompson. Internet X.509 public key infrastructure proxy certificate profile. *The Internet Engineering Task Force (IETF)*, RFC 3820, June 2004.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Can Intelligent Optimisation Techniques Improve Computing Job Scheduling In A Grid Environment? Review, Problem and Proposal

Wei Huang[1], Tim French[1,2], Carsten Maple[1], Nik Bessis[1]

[1] Department of Computing and Information Systems, University of Bedfordshire, Park Square, Luton, LU1 3JU, UK (Corresponding author: wei.huang@beds.ac.uk)
[2] Applied Semiotics with Informatics Laboratory, Informatics Research Group, School of Systems Engineering, University of Reading, Whitenights, RG6 6AH, UK

Abstract

In the existing Grid scheduling literature, the reported methods and strategies are mostly related to high-level schedulers such as global schedulers, external schedulers, data schedulers, and cluster schedulers. Although a number of these have previously considered job scheduling, thus far only relatively simple queue-based policies such as First In First Out (FIFO) have been considered for local job scheduling within Grid contexts. Our initial research shows that it is worth investigating the potential impact on the performance of the Grid when intelligent optimisation techniques are applied to local scheduling policies. The research problem is defined, and a basic research methodology with a detailed roadmap is presented. This paper forms a proposal with the intention of exchanging ideas and seeking potential collaborators.

1. Introduction

In recent years, there has been increasingly interest in using network-based resources for large scale data-intensive computation problems. These problems, usually found in a number of disciplines such as high-energy physics, astronomy and bioinformatics, involve loosely coupled computing jobs and geographically distributed resources e.g. supercomputing powers and large datasets. Within this context, the Grid computing paradigm originated as a new infrastructure to address these problems and is now an established technology for large scale resource sharing and distributed integration within both science and industry setting [1]. As one of the most important future trends, the importance of Grid is widely recognised and Grid-related projects are heavily funded world wide, e.g. US Globus, UK e-Science, and EU FP6.

Effective computation and data scheduling is rapidly becoming one of the main challenges in Grid computing, and is seen as being vital for its success. Different strategies have been proposed for effective job and data scheduling for such complex systems [2-5]. Ranganatham and Foster in [2] proposed a generic Grid scheduling architecture in which the scheduling logic is encapsulated in three modules: External Scheduler (ES), Local Scheduler (LS) and Dataset Scheduler (DS). Each user submits jobs to an external scheduler that decides to which remote site these jobs are allocated. The local scheduler at a site decides how to schedule all pending jobs based on available resources. The dataset scheduler at each site keeps track of the popularity of each dataset requested and it will replicate popular datasets to remote sites depending on a number of strategies. A total of twenty different ES and DS algorithm combinations were proposed and evaluated. However, in order to simplify the study, FIFO (first in first out) was used as a local scheduling policy. Yarmolenko et al. [6] also indicated that traditionally the enactment of jobs on parallel computing resources have been based on queue-based scheduling systems, namely “run the job when it gets to the head of the queue”.

This paper reports upon the state-of-the-art literature review on the methodologies used for Grid scheduling, particularly those for local scheduling policies in a Grid environment. The rest of the paper is structured as follows. Section 2 reviews related work in Grid scheduling. In Section 3, the problem of local scheduling policies is identified and the research objectives are presented. A basic research methodology containing a detailed research roadmap is given in Section 4 and Section 5 concludes the paper.

2. Related Work Review

A variety of factors need to be considered for the effective scheduling of resources in Grid environments, e.g. resource utilization, response time, global and local allocation policies and scalability. Ranganatham and Foster [2] indicate that effective scheduling in a Grid system is complex. The large amount of input data needed for each computing job would suggest that job scheduling strategies should ideally take data locality into account. Considering the large number of users and resources that a Grid system supports, decentralized strategies may perform better than centralized strategies. Thus, an important aspect for Grid scheduling is to allow local scheduling policies. Yarmolenko et al. [6] indicate that it was only recently that more flexible approaches were used and Service Level Agreements (SLAs) could effectively be used for job farming. In their paper, they presented their work on an UK e-Science project that investigated the effects of SLA based approaches for computing job scheduling. A coordinator based architecture and evaluation of different policies for the negotiation of SLAs between the coordinator and the resources were considered. The work showed that the use of information available in the SLAs agreed by the User might fundamentally alter the behaviour of the coordinator and hence the performance of the overall system. However, these effects need to be understood in more detail before new designs can take advantage of the possibilities offered by SLAs.

In another recent UK e-Science project, Palmer and Mitrani [7] considered optimising the tree structure of large-scale Grids containing many processors. They argued that a "flat" structure, where only one single level master node (i.e. a global scheduler) controlled all processors and decided where incoming jobs should be executed, was not always efficient as the master node could become easily overloaded when demand was high. A tree structure involving a hierarchy of master nodes (i.e. global and local schedulers) could control subsets of processors so as to avoid bottleneck problems but might introduce additional processing and transfer delays. A simple heuristic approach was adopted in their paper for the dynamic reconfiguration of the tree structure as load changed. It was shown through numerical experiment that, for a given set of parameters and job distribution policy, there was an optimal tree structure that minimized the overall average response time.

Kubicek et al [8] also proposed an architecture that allowed the dynamic reconfiguration of servers to process incoming jobs by switching servers between conceptual pools. A number of heuristic policies have been used to make optimal switching decisions, and a prototype system was developed to demonstrate these concepts.

Thomas et al. [9] investigated the performance effects of a delay in propagating information concerning computing node failure. In Grid computing environments, scheduling will generally be performed by global or external schedulers remotely from computing nodes. However, if a node has failed it is in no position to communicate its state and mechanisms should be applied to let the schedulers know this. The authors indicate that none of the current studies adequately deal with the consequence of failure at the resource level and very few services have been constructed with a fault tolerant perspective.

Cawood et al. [10] developed two fully Globus-enabled Grid scheduling tools, TOG (Transfer-queue Over Globus) and JOSH (JOb Scheduling Hierarchically), based on the Grid Engine and the Globus Toolkit (GT). Grid Engine is an open source distributed resource management system for computing resources within an organisation and the Globus Toolkit is an API for connecting distributed resources among different organizations. TOG has the potential to integrate Grid Engine V5.3 and Globus Toolkit V2.2 to allow access to remote resources, and JOSH used the Golbus Toolkit V3's Managed Job Service (MJS) to run the job submission and termination scripts on behalf of the client user. However, the authors also indicate that due to the performance and robustness concern of the GT V3.0, a number of organisations have reluctantly decided not to consider JOSH for deployment. JOSH take-up might be further affected by the switch to WS-RF in GT 4.0.

3. Problem Definition

In the existing Grid scheduling literature, the reported methods and strategies are mostly related to high-level schedulers such as global schedulers, external schedulers, data schedulers, and cluster schedulers. Although a number of these have previously considered job scheduling, thus far only relatively simple queue-based policies such as First In First Out (FIFO) [1, 6] have been considered for local job scheduling within Grid contexts. An interesting question arises based on the current

state-of-the-art Grid literature: what happens if more intelligent techniques are adopted for local scheduling within a Grid environment? Furthermore, will these techniques help to improve the efficiency of job scheduling and resources management for the whole Grid? For example, due to the relatively slower data transfer rate and frequent data transfer latency between geographically distributed Grid services, local scheduling policies need to consider factors such as jobs' priority, temporal constraints, jobs' waiting policies, datasets limitation, finite capacity of storage for job queues and data, and resource failure. Resource trust expectations also need to be considered and confidence levels can potentially become one of the criteria used to schedule and/or reschedule resources [11-12]. Under these circumstances, a queue-based scheduling policy is simply not good enough and an intelligent method will be more efficient.

This research aims to investigate the impact on the performance of a Grid system when intelligent optimisation techniques are adopted by local schedulers within a generic Grid scheduling architecture. After the completion of the planned research, it will become clear which local scheduling policies and their combination are efficient so that they can improve the performance of the whole Grid computing.

The specific research objectives are namely:

1. To identify a variety of factors that local schedulers need to consider within a general Grid scheduling architecture;
2. To develop intelligent scheduling models and algorithms for computation and data scheduling within a Grid system;
3. To develop a simulator that can be used to evaluate the performance of a Grid system adopting one or more intelligent techniques for scheduling;
4. To demonstrate the efficiency of the developed models and methodologies through simulated Grid environments.

These research objectives are important to the future study of the Grid. The Grid paradigm originated as a new computing infrastructure for data-intensive computing problems and is now an established technology for the sharing of large scale geographically distributed resources. Achieving these objectives will make the resource allocation more efficient within a Grid.

4. Proposed Research Methodology

The project will follow a systematic approach as follows:

1. State-of-the-art review. Several factors are assumed to be considered for Grid computing e.g. jobs' priority, temporal constraints, jobs' waiting policies, datasets and storage limitation. The first step of the project will be aimed at verifying these assumptions and matching them to current state-of-the-art.
2. Pilot scenarios development. Key characteristics will be identified for the Grid global and local scheduling and a number of pilot scenarios will be developed for assessing the impacts of local policies on the performance of different strategies of external scheduler (ES) and dataset scheduler (DS).
3. Grid system modelling. The Grid system will be modelled as a network of computing sites distributed geographically, each comprising a number of processors and a limited amount of storage. It is assumed that a number of users, each one associated with a particular site, submit jobs. Each job requires specific computation and dataset resources to be available, which may not reside on the site where the job is submitted. The general Grid scheduling architecture proposed in [2] will be adopted.
4. Modelling of scheduling problems. Novel constraint-based scheduling models for local schedulers within a Grid system will be proposed. The problem can be considered as constraint satisfaction problems (CSPs) where a set of jobs, a set of processors and a limited amount of data storage are given; each job requires a number of processors and an amount of storage for a certain time, satisfying a set of constraints such as jobs' priority, time-bound constraints etc, and the objective is to maximize the efficiency of a computing site i.e. to schedule as many jobs as possible within a deadline.
5. Development of scheduling

algorithms. A number of intelligent optimisation techniques including constraint programming (CP) will be considered to solve scheduling problems with a large number of constraints quickly. Specific scheduling algorithms and approaches for local schedulers within a Grid environment will be developed.

6. Simulator development. The project plans to evaluate the performance of developed models and methodologies by simulation that can allow the test of a wide range of scenarios. Currently, a number of Grid simulators, such as ChicagoSim, GridSim, SimGrid, and OptorSim, are available. A specific discrete event simulator for the project will be developed on top of one of available Grid simulators.
7. Verification programme. The developed models and methodologies will be integrated within the general Grid scheduling architecture [2] and demonstrated through a simulator-based Grid environment. The verification will be based on integrating different combinations of scheduling strategies of external scheduler (ES) and dataset scheduler (DS) with the developed scheduling approaches of local scheduler (LS) within a common environment. The effect on the performance of different ES and DS scheduling algorithms by developed models and methodologies will be investigated.

5. Conclusions and Future Work

This paper has reviewed state-of-the-art literature in Grid scheduling and has identified some potential problems in local scheduling policies within Grid systems. The initial literature research has established that it is worth investigating the impact of scheduling Grid computing by using intelligent optimisation techniques within local scheduling policies so as to deal with more complex constraints in job scheduling and resource management. A basic research methodology with a detailed roadmap for future research is proposed. Constraint programming techniques will be used in this investigation and other intelligent optimisation techniques will also be considered. The viabilities of the proposed scheduling models and algorithms will be evaluated through simulated Grid environments.

References:

[1] I. Foster, C. Kesselman, 1999. The Grid: Blueprint for a new computing infrastructure. Morgan Kaufmann, San Mateo.

[2] K. Ranganathan and I. Foster, 2003. Simulation studies of computation and data scheduling algorithms for data grids. Journal of Grid Computing, 1(1), 53-62.

[3] M. Caramia, S. Giordani and A. Iovanella, 2004. Grid scheduling by on-line rectangle packing. Networks, 44(2), 106-119.

[4] Y. Gao, H. Rong and J. Huang, 2005. Adaptive grid job scheduling with genetic algorithms. Future Generation Computer Systems, 21, 151-161.

[5] C. Weng and X. Lu, 2005. Heuristic scheduling for bag-of-tasks application in combination with QoS in the computational grid. Future Generation Computer Systems, 21, 271-280.

[6] V. Yarmolenko, R. Sakellariou, D. Ouelhadj, J. M. Garibaldi, 2005. SLA Based Job Scheduling: A Case Study on Policies for Negotiation with Resources. Proceedings of the UK e-Science All Hands Meetings, Nottingham, 2005.

[7] J. Palmer, I. Mitrani, 2005. Optimal Tree Structures for Large-Scale Grids. Proceedings of the UK e-Science All Hands Meetings, Nottingham, 2004.

[8] C. Kubicek, M. Fisher, P. McKee, R. Smith, 2004. Dynamic Allocation of Servers to Jobs in a Grid Hosting Environment. Proceedings of the UK e-Science All Hands Meetings, 2004.

[9] N. Thomas, J. Bradley, W. Knottenbelt, 2004. Performance of A Semi Blind Service Scheduler. Proceedings of the UK e-Science All Hands Meetings, Nottingham, 2004.

[10] G. Cawood, T. Seed, R. Abrol, T. Sloan, 2004. TGO & JOSH: Grid Scheduling with Grid Engine & Globus. Proceedings of the UK e-Science All Hands Meetings, Nottingham, 2004.

[11] S. Ramchurn, D. Huynh and N. Jennings, 2004. Trust in multi-agent systems. The Knowledge Engineering Review, Vol. 19:1, 1-25.

[12] T. French and W. Huang, (2005). Grid enabled collaborative computing: Is trust the hardest issue to address?. Proceedings of the 11th Chinese Automation and Computing Society Conference in the UK, pp131-136, ISBN 0 9533890 8 1, Sheffield, England, 10 September 2005.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Preserving Scientific Data with *XMLArch* *

Peter Buneman James Cheney **Carwyn Edwards** Irini Fundulaki

{opb,jcheney,cedward1,efountou}@inf.ed.ac.uk

Abstract

Scientific databases are continuously updated either by adding new data or by deleting and/or modifying existing data. It is fairly obvious that it is crucial to preserve historic versions of these databases so that researchers can access previous findings to verify results or compare old data against new data and new theories. In this poster we will present XMLArch that is used in archiving scientific data stored in relational or XML databases. XMLArch builds upon and extends previous archiving techniques for hierarchical scientific data.

1 Introduction

The Web is now the most important means of publishing scientific information. In particular, an increasing number of scientific databases are published on the Web, allowing scientists to exchange their research results. Most of these databases are continuously updated either by adding new data or by deleting and/or modifying existing data. In order to preserve the scientific record it is crucial to keep versions of these databases. Researchers should be able to validate previous findings and to compare old data against new data and new theories. It is reported in [3] that archiving is a ubiquitous problem. Even databases that record ostensibly fixed data, such as the results of experiments or simulations have associated metadata, such as classification information and annotation, and this metadata is almost always subject to change.

In this paper we report on progress in constructing *XMLArch*, a generic system for archiving scientific data represented in XML. Based on this we have constructed a simple tool that one can simply point at a relational database. It extracts the data into a default XML format with associated key information and then incorporates the XML into an incremental archive. Not only does this preserve successive versions of the database, it preserves them in a fashion that is independent of any specific relational database management system and makes possible temporal queries on data.

We use the IUPHAR pharmaceutical database [8] as a concrete example to discuss the problems arising with archiving real scientific data. IUPHAR (International Union of Pharmacology) was founded in 1959 and one of its main objectives is to foster international cooperation in pharmacology by promoting cooperation between societies that represent pharmacology and related disciplines throughout the world.

The IUPHAR database is continuously updated and it is published two to three times a year. Archiving the different versions of the database is crucial since it is widely cited by researchers in their work who need to access its different versions. The IUPHAR curators keep versions of the database as database dumps (complete snapshots) and only the latest version of the database is live. Such snapshots unfortunately cannot easily be queried and it is extremely cumbersome for one to access historic information, for example the change history for a given receptor. It is also evident that there is a significant storage overhead in keeping all the versions of the database.

Apart from this brute-force approach in archiving scientific data, other approaches based on storing the differences (or deltas) between the different versions of text documents are also common. These approaches, that are based on line-diff algorithms, clearly conserve space and scale well. However, retrieving an older version might involve either undoing or applying many deltas. In a similar manner, finding how an element has evolved is also a problem and may require complicated reasoning using the recorded deltas. This is because the current approaches based on differences do not preserve the structure of the database, hence the identity of objects is lost.

In this poster we will demonstrate *XMLArch* that is used in archiving scientific data stored in relational or XML databases. The approach has been presented in [2] and is based on the idea of merging all the database versions into a single archive. It takes advantage of the hierarchi-

* This project has been supported in the Digital Curation Centre, which is funded in part by the EPSRC eScience core programme.

cal structure of the data and leverages the strong key structures which are common in the design of scientific datasets. We will also present the benefits of combining archiving as done in *XMLArch* and XML compression, using for the latter the state of the art techniques as surveyed in [5].

2 Archiving Scientific Data

In this section we give a high level overview of the *XMLArch* archiver. As already mentioned, we choose the archiving approach presented in [2]. This work dealt with archiving scientific data that is stored in some hierarchical data format (such as XML). In the case of relational data we first need to extract the relational data into a hierarchical format, in this case to XML. To do this we built a simple relational database extraction tool, the output of which is then passed into the archiving tool itself and optionally on to a compressor.

The idea for the initial relational database extractor was that it should not need to know anything at all about the semantics of the domain data. To facilitate this the extractor uses a simple schema-agnostic XML format to represent the extracted relational data, making the implementation fairly portable across different databases and domains. The only input required is access to the relational database itself. The motivation for this simplistic approach being that often the person archiving a dataset has little or no prior experience with the data (e.g. a Systems Administrator).

This *point and extract* behavior makes the extractor behave very much like conventional relational database backup tools. The eventual aim is that as much as possible of the relational information in the database will be preserved along with the XML snapshot. The theory being that as long as the data is preserved along with some form of structural description, possibly no more than a textual description of the relations, then someone in the future will be able to reconstruct the relations around the data. Obviously if we can store as much of the schema as possible in a standard form this reduces the work needed to recreate it.

One last aim was that the data extractor should not attempt to do too much. Tools already exist from virtually every relational database vendor to extract data into XML. The extractor component of this project should be seen as a simplest possible tool to use in order to get the data out of the database in the absence of other more suitable options.

Once the relational data is extracted it is archived by the XML archiver submodule of *XMLArch* using an approach based on the one introduced in [2]. The archiving techniques presented in that work stem from the requirements and properties of scientific databases: first of all, much scientific data is kept either in well-organized hierarchical data formats, or it has an inherent hierarchical structure; second, this hierarchically structured data usually has a key structure which is explicit in the design. The key structure provides a canonical identification for every part of the document, and it is exactly this structure that is the basis of the technique in [2].

In [2] the idea behind archiving is based on:

1. *identifying the correspondence and changes between two given versions based on keys* and
2. *merging the different versions using these keys in one archive*.

Identifying correspondences between versions using keys is different from the *diff-based* ([7]) approaches used in most of the existing tools. *Diff-based* approaches are based on minimum edit distances between raw text lines and use no information about the structure of the underlying data. Our key based approach on the other hand attempts to unify objects in different data versions based on their keyed identity. In this way, the archive can not only preserve the semantic continuity between the elements but also efficiently support queries on the history of those elements.

The merging of different versions into one archive is again different from the *diff-based* approaches that store the deltas from version to version independently. Occurrences of elements are identified using the key structure and stored only once in the final archive. A sequence of numbers (*timestamp*) is used to record the sequence of versions in which an element appears. This timestamp is conceptually stored with every element in the hierarchy. Taking advantage of the hierarchical nature of the data, the timestamp in an element is stored only when it is different from that of its parent. In this way, a fair amount of space overhead is avoided by inheriting timestamps.

The final archive is stored as an XML document. XML was chosen as the archive format for a number of reasons: for one thing the hierarchical models used in many biological databases are very similar to the XML data model. In addition, XML is currently the most prevalent textual format for hierarchical data. Because of this there are a significant number of commercial and open source tools that are readily available to manipulate XML. Given that one of the most important considerations when archiving is to make sure that the data is retrievable at a later date, using a widespread, *human readable* text-based format would seem to increase the likelihood that the data will be accessible in the future.

The final XML archive is then optionally passed into standard text or XML compression tools to further reduce the storage requirements of the archive. As the same principles for archiving as noted in [2] are used by the XML archiver, the same benefits in terms of storage overhead reductions noted in that work apply. As was shown in [2] the resulting output is often particularly well suited to XML specific compressors such as XMill [9]. This is unsurprising given the hierarchical models underlying the source databases and is supported by the findings of [4].

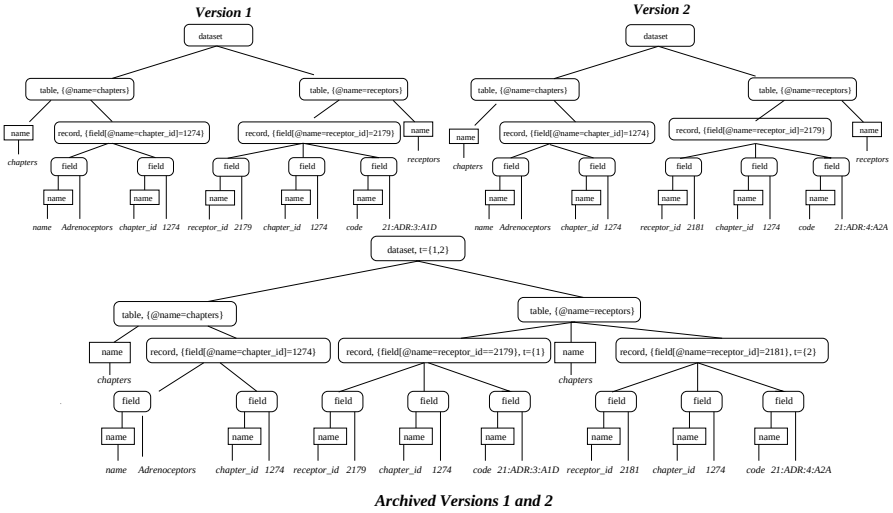


Figure 1. Archived Versions of the IUPHAR database

3 An Example: Archiving the IUPHAR database

In this section we present an example to demonstrate how *XMLArch* is used to archive IUPHAR relational data. To present our approach we use a simple example of IUPHAR data.

We consider tables chapters and receptors shown below (the primary keys for all tables are underlined). The first stores the receptor families where each family has a $chapter\_id$ (primary key) and a $name$. Table receptors stores the $name$ and $code$ of receptors. Attribute $receptor\_id$ is the primary key for the table and attribute $chapter\_id$ indicates the receptor family to which the receptor belongs to.

Source relational schema R:

chapters($\underline{chapter\_id}$, $name$)
receptors($\underline{receptor\_id}$, $chapter\_id$, $name$, $code$)

The XML DTD to which this data is published is shown below:

```
1. <!ELEMENT dataset (table*)>
2. <!ELEMENT table (record*)>
3. <!ATTLIST table name #PCDATA>
4. <!ELEMENT record (field+)>
5. <!ELEMENT field #PCDATA>
6. <!ATTLIST field name #CDATA
7.           keyfield (true|false) 'false'>
```

Element `table` stores information about a relational table. Attribute `name` records the name of the table (line 3). Each `table` element has one or more `record` elements (line 2) where such an element is defined for each tuple of the corresponding relational table. A `record` element has one or more `field` subelements (line 4). A `field` element is defined for an attribute of the relational table with XML attribute `name` to record the name of the relational attribute (line 6). Attribute `keyfield` records whether the attribute participates in the primary key of the relational table or not (line 7).

In addition to this XML data, we also need a way to identify the *XML elements* in the extracted XML document. This is done by using the notion of *XML keys* introduced in [1]. In this work an XML element is *uniquely identified* by the values of a subset of its *descendant* elements.

For example, in our case a `table` element is uniquely identified by its `name` attribute. A `record` element within the `table` element defined for the relational table chapters (i.e., the `table` element with value *chapters* for its `name` attribute) is uniquely identified by the value of its `field` subelement that corresponds to the relational attribute $chapter\_id$. In a similar manner, a `record` element within a `table` element that corresponds to the receptors relational table is uniquely identified by the value of its `field` subelement defined for the $receptor\_id$ relational attribute. This key information is defined by means of XPath [6] expressions as advocated in [1]. In *XMLArch* the annotator component, a sub component of the archiver, records with each keyed element its XML key and value whenever applicable (e.g. elements `table` and `record`). These key annotations are used during the merging of a new version of the database with the archived version to unify element identities.

We show in the upper part of Figure 1 two versions of the database. The difference between Version 1 and Version 2 is that receptor with receptor_id equal to 2179 has been deleted and receptor with receptor_id equal to 2181 has been added.

The archived XML document is also shown in Figure 1. One can observe that elements that exist in both versions are stored only once (e.g., the `table` elements and the `dataset` element). Notice that there is a timestamp ($t = \{1,2\}$) associated with the `dataset` element that indicates that this element (as well as all its descendants that do not have a timestamp) are present in both versions. Observe that the `record`

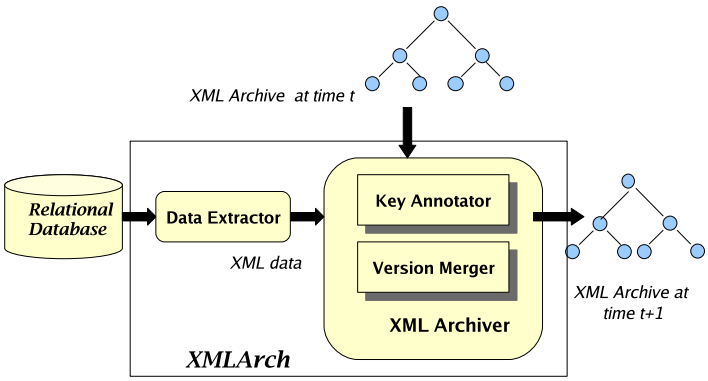


Figure 2. *XMLArch* **Architecture**

element deleted in the second version has a timestamp equal to "1" ($t = \{1\}$) whereas the `record` element added in the second version has timestamp equal to "2" ($t=\{2\}$).

4 System Architecture

The architecture of the system is shown in Figure 2. The **Data Extractor** is responsible for extracting the relational data into the XML format discussed previously. This component reads the schema of the database (tables and constraints such as primary keys) and the instances (i.e., tuples) and exports the database in XML.

The exported data is then passed to the **XML Archiver** that is responsible for creating the XML archive. This module consists of the **Key Annotator** and **Version Merger** submodules. The former is responsible for annotating each XML element with its key, and the latter for merging the latest XML archive (archive at time `t`) with the new version to produce the archive at time `t+1`.

References

[1] P. Buneman, S. Davidson, W. Fan, C. Hara, and W. Tan. Keys for XML. In *WWW*, 2001.

[2] P. Buneman, S. Khanna, K. Tajima, and W. C. Tan. Archiving Scientific Data. *TODS*, 2004.

[3] The WWW Virtual Library of Cell Biology. http://vlib.org/Science/Cell_Biology/databases.shtml.

[4] J. Cheney. Compressing XML with Multiplexed Hierarchical Models. In *In Proc. IEEE Data Compression Conference (DCC)*, 2001.

[5] J. Cheney. An Empirical Evaluation of Simple DTD-Conscious Compression Techniques. In *WebDB*, 2005.

[6] J. Clark and Steve DeRose. XML Path Language (XPath) 1.0. W3C Recommendation, 1999. http://www.w3c.org/TR/xpath.

[7] J. W. Hunt and M. D. McIlroy. An algorithm for differential file comparison. Technical Report CSTR #41, Bell Telephone Laboratories, 1976.

[8] IUPHAR. Receptor Database. `http://www.iuphar-db.org`.

[9] H. Liefke and D. Suciu. XMill: an Efficient Compressor for XML Data. In *SIGMOD*, pages 153–164, 2000.

Proceedings of the UK e-Science All Hands Meeting 2006 © NeSC 2006 ISBN 0-9553988-0-0

FEA of Slope Failures with a Stochastic Distribution of Soil Properties Developed and Run on the Grid

William Spencer[1], Joanna Leng[2], Mike Pettipher[2]

[1] School of Mechanical Aerospace and Civil Engineering, University of Manchester
[2] Manchester Computing, University of Manchester

Abstract

This paper presents a case study about how one user developed and ran codes on a grid, in this case in the form of the National Grid Service (NGS). The user had access to the core components of the NGS which consisted of four clusters which are configured, unlike most other U.K. academic hpc services, to be used primarily through grid technologies, in this case globus. This account includes how this code was parallelised, its performance and the issues involved in selecting and using the NGS for the development and running of these codes. A general understanding of the application area, computational geotechnical engineering, and the performance issues of these codes are required to make this clear.

1. Introduction

The user is investigating the stochastic modelling of heterogeneous soils in geotechnical engineering using the Finite Element Analysis (FEA) method. Here thousands of realisations of soil properties are generated to match statistical characteristics of real soil, so that margins of design reliability can be assessed. The user wished to understand what performance benefits they could gain from parallelising their code.

To do this the user needed to develop a parallel version of the code. The production grid service philosophy of the NGS seemed appropriate to run the multiple realisations necessary. To do this the user had to test and run the code through grid technologies, which in this case meant globus.

2. Scientific Objectives

Engineers characterise material property, such as the strength of steel, by a single value, in order to simplify subsequent calculations. By choosing a single value, any inherent variation in the material is ignored. In soil, widely differing properties are seen over a small spatial distance, invalidating this assumption.

In this case the model is a 3D representation of a soil slope or embankment. The slope fails when a volume of soil breaks away from the rest of the slope and moves en-masse downhill due to gravity, Figure 1. This process is modelled using a FEA with an elastic-perfectly plastic Tresca model to simulate a clay soil.

Previous studies in this field, e.g. [5], have been limited to 2D analysis with modest scope. These 2D models are flawed in their representation of material variability and failure modes, thus the novel extension to 3D provides a much greater understanding of the problem.

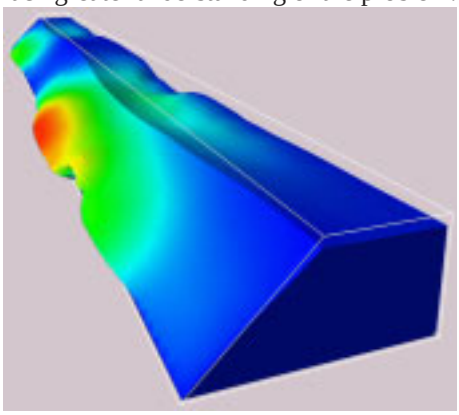


Figure 1: Example of random slope failure contours of displacement

Stochastic analysis was performed using the Monte Carlo framework to take account of spatial variation. A spatially correlated random field of property values is generated using the LAS method [2]. This mimics the natural variability of shear strength within the ground; this is mapped onto a cubiodal FE mesh. Figure 2 shows a 3D random field, typical of measured values for natural soils [1]. FEA is then performed in which the slope is progressively loaded, until it fails. This process is then repeated with a different random field for each realisation. After many realisations the results are collated allowing the probability of failure to be derived for any slope loading.

A further limitation is placed on the mesh resolution in order to preserve accuracy, the maximum element size is 0.5m cubed. 500 realisations were necessary to gain an accurate understanding of the slope failures reliability.

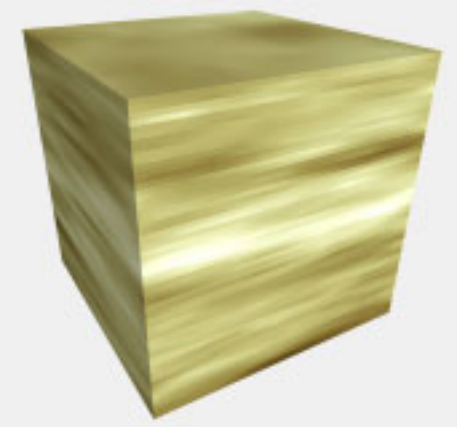


Figure 2: 3D random field

3. Strategy for Parallelisation

Two approaches were investigated, one that uses a serial solver, but achieves parallelism by task farming realisations to different processors; and the other that uses a parallel solver and task farming. Both codes were adapted from their original forms [4].

Once developed both codes were analysed to discover which would be most appropriate for full scale testing. A serial version, using a direct solver, was developed on a desktop platform. The second, a simple parallel version that uses an iterative element by element solver, was developed and optimised on a local HPC facility, comprising SGI Onyx 300 platform with 32 MIPS based processors.

The limiting factor preventing further work on the Onyx was time to solution, with serial solutions taking six times longer than on the user's desktop. In order to allow larger analyses to be conducted 100000 CPU hour were applied for, and allocated, on the NGS. The NGS consists of 4 clusters of Pentium 3 machines, each processor running approximately 1.5 times faster than the user's desktop and 10 times faster than the Onyx processors.

4. Performance

This is a plasticity problem that uses many loading iterations to distribute stresses and update the plasticity of elements. The visco-plastic strain method used requires hundreds of iterations but no updates to the stiffness matrix. This allows the time consuming factorisation of the stiffness matrix to be decoupled from the loading iterations. The iterative solver uses the same stress return method, however in this solver the factorisation and the loading are inherently linked. Other stress return methods are not available for the Tresca soil model used.

4.1 Comparison of direct and iterative solvers

It is the normal expectation that the iterative solver would outperform the serial direct solver in a parallel environment. Indeed the iterative solver showed good speedup when run on increasing numbers of processors, as expected by Smith [4]. The plasticity analysis discussed here does not fit this generalisation. Consideration of the mathematical procedures adopted in this iterative solver and in the direct solver lead to the observation that the direct solver takes advantage of the unchanging mesh.

Table I compares timings for codes with both solvers, the iterative and the direct, for one realisation only. The code with the direct solver was run on 1 processor while the code with the iterative solver was run on 8 processors. The table demonstrates the efficiency of the direct solver for this problem in the ratio of timings for the two solvers in terms of wall time and so demonstrating the algorithmic efficiency of the direct solver for this particular analysis.

When the use of task farming for multiple realisations is studied, the direct solver becomes even more competitive; assuming efficient task farming over 8 processors, then this is demonstrated by the ratio of CPU time. The serial solver strategy is an order of magnitude faster than the parallel solver.

The minimum desired mesh depth for this problem is 60 elements, with 500 realisations and 16 sets of parameters being needed for a minimal analysis. The user found that feasibly only 32 processors were available for use on the NGS. Based on extrapolation of Table I a rough estimate of wall clock time can be calculated; task-farming with the direct solver requires 5.8 days while task-farming with the iterative solver requires 57.9 days. Clearly the direct solver combined with running realisations in parallel is the only reasonable choice.

4.2 Memory requirements

The only major drawback is that the direct solver consumes much more memory than the iterative solver. The resources for the direct solver are greater as it needs to form and save the entire stiffness matrix. The required memory increases with the number of degrees of freedom squared. Whereas the iterative solver has a total memory requirement that grows with

Number of Elements in y Direction (length of slope)	Direct Solver on 1 processor (column A)	Parallel Solver on 8 processors (column B)	Wall Time	CPU Time
	Time (s) per Realisation		Ratio of B/A	
1	27.7	111.7	4.0	32.3
3	110.9	245.0	2.2	17.7
5	200.9	392.6	2.0	15.6
15	592.4	982.4	1.7	13.3
30	1092.5	1601.1	1.5	11.7

Table I: Comparison of direct and iterative solver run times.

the number of degrees of freedom divided by the number of processors used.

Thus the direct solver is limited to the amount of memory locally available to the CPU, which on the two main NGS nodes is 1 Gb. This gives an absolute limit to the number of elements that can be solved; In this case it is fortunate that the maximum size of mesh is large enough to give a meaningful solution.

The results in Table I, show that the iterative solver is increasingly competitive the larger the problem gets. Combined with the memory limitation of the direct solver a further increase in mesh size could only be achieved using the iterative solver.

5. Experimentation

The validity of the final code was proven by comparing it with the results of the previously studied 2D version [5] and those of well established deterministic analytical results. In both cases the results compared very favourably, providing a check on both the 3D FEA code and the 3D random field generator.

Further to the validation, full scale analyses are currently being undertaken. Preliminary results [6] show that use of the more realistic 3D model has a significant effect on the reliability of the slope. These results have interesting implications for the design of safe and efficient slopes, showing that a 'safe' slope designed by traditional methods can either be very conservative or risky, if the effects of soil variability are not considered.

6. Suitability and Use of the National Grid Service (NGS)

The use of commodity clusters is ideal for this code's final form, as its fundamental nature does not require the ultra fast inter-processor communication or shared memory of some proprietary hardware. The very large memory requirements of the direct solver made the individual per processor memory limit the constraining factor.. For the alternate iterative solver version high speed interconnections would go some way to speeding it up.

The NGS is configured to be used through globus. In this work globus was used for several purposes from interactive login, to file transfer and job submission.

The user is based as the School of Engineering at the University of Manchester and it is worth noting that this school has a policy of only supporting Microsoft windows on their local desktop machines and globus is not part of the routine configuration. Globus was not installed locally but instead the user started a session on a local Onyx and started their globus session from there.

Once the user had his code working on one of the NGS nodes he transferred and compiled a copy on each of the other nodes. Job submission was performed using each of these executables. The code is designed to run in a particular environment with set directories and input and out put files. The user developed a number of scripts to automate the set up of this configuration and these were run by hand before a job was run. With time and confidence the user could completely automate the process.

The jobs were submitted from the Onyx using scripts to write and execute pbs, these were configured to provide a wide range of job submission types and then reused or adapted ad hoc. The initial scripts were quite simple with just a `globus-job-run` command. These saved the user typing out complex commands. More sophisticated scripts were developed as the user became confident. These scripts set environment variables and used the Resource Allocation Language file for job submission. The user monitored the load on all the nodes so that he could submit jobs on the node with the lowest load.

A small amount of work was needed to get the codes that had previously been running on the Onyx to run on the NGS machines. The compiler on the NGS nodes was stricter in its implementation of standards. The method adopted for editing the code was to do so on a

desktop windows machine in a FORTRAN editor and then transfer the file via the local Onyx to the NGS node, whence it was compiled. This did take some extra time in transferring files to and fro but in general the reduction in execution time from running tests on NGS more than made up for it.

Debugging was achieved through printing output to file and by visualization when the code ran to completion, but incorrectly. While totalview is available on the nodes the user was not aware of this tools functionality or how to use this profiling tool.

Overall developing the code on the grid was no more painful than in any other environment, the only downside being the lack of processor availability when the service is busy; either requiring the use of a different NGS node or the local Onyx.

7. Discussion and Conclusions

It should be noted that prior to this project on the NGS this user had no experience or understanding of e-Science and the grid. He was not familiar with hpc and had little practice in using HPC services. Initially it was daunting to use a service like the NGS where the policies and practices were different to the user's local HPC service. The user took some time and support to learn how to get the best out of the service but in the end was happy with both the service and the computational results.

The main value of using the NGS for this user was to allow the execution of code requiring large amounts of CPU time. Large volumes of CPU time for in house machines are often hard to come by because they are in high demand. At present the NGS is moderately loaded, and has powerful computers, allowing such large analyses to be run in a timely fashion. Generally a run requiring 32 CPU's was started within 24 hours. It also (usually) allowed virtually on demand access to run small debug or test programs, most useful in code development.

The NGS has been in full production since September 2004 and currently has over 300 active users. This user applied for resources near the beginning of the service, autumn 2004. The loading of the service has increased steadily and with this the monitoring of the use of resources has become more critical [3]. As the number of users increases further and the resources become scarcer it is expected that the policy of the NGS will develop. Given the very large allocation of time given to this user (100000 hours), this has not been of severe detriment, but may constrain other users with smaller allocations.

In its present form the ideal solution to filling the desired number of CPU hours would be to harness the spare CPU time of the university's public clusters via distributed grid application such as BOINC [7] or Condor [8]. Short of this the use of the NGS provides a ready to run and largely trouble free resource, with good support services.

The future of this particular application is no doubt in a fully parallel implementation, with the iterative solver. This is the only computational approach that will deal with the demands of the science, which will require the analysis of higher resolution and longer slopes. Little optimisation or profiling was used on any of the codes. It is expected that improvements to efficiency and speed would be introduced particularly to the iterative solver version. Further development of the tangent stiffness method making it applicable to this problem would dramatically improve the performance of the iterative solver, at the expense of considerable research time.

Acknowledgment

The authors would like to acknowledge the use of the UK National Grid Service in carrying out this work.

References

[1] K. Hyunki, "Spatial Variability in Soils: Stiffness and Strength", *PhD thesis*, Georgia Institute of Technology, August 2005, pp 15-16.

[2] G.A. Fenton and E.H. Vanmarcke, "Simulation of random fields via local average subdivision." *J. of Engineering Mechanics, ASCE*, 116(8), Aug 1990, pp 1733-1749.

[3] NGS (National Grid Service); http://www.ngs.ac.uk/, last accessed 21/2/06.

[4] I.M. Smith and D.V. Griffiths, "Programming the finite element method" third edition, John Wiley & Son, Nov 1999.

[5] M.A. Hicks and K. Samy, "Reliability-based characteristic values: a stochastic approach to Eurocode 7", Ground Engineering, Dec 2002, pp 30-34.

[6] W. Spencer and M.A. Hicks, "3D stochastic modelling of long soil slopes", 14th ACME conference, Belfast, April 2006, pp 119-122.

[7] Berkley open infrastructure for network computing; http://boinc.berkeley.edu/, last accessed 2/4/06.

[8] Condor for creating computational grids; http://www.cs.wisc.edu/pkilab/condor/, last accessed 4/7/06

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

On Building Trusted Digital Preservation Repositories

Reagan W. Moore, Arcot Rajasekar, Michael Wan, Wayne Schroeder, Richard Marciano

San Diego Supercomputer Center

Abstract

Trusted digital repository audit checklists are now being developed, based on assessments of organizational infrastructure, repository functions, community use, and technical infrastructure. These assessments can be expressed as rules that are applied on state information that define the criteria for trustworthiness. This paper maps the rules to the mechanisms that are needed in a trusted digital repository to minimize risk of both data and state information loss. The required mechanisms have been developed within the Storage Resource Broker data grid technology, and their use is illustrated on existing preservation repository projects.

1. Introduction

A trusted digital repository uses explicitly defined policies to manage records. These policies can be validated against criteria published by the RLG and NARA in "An Audit Checklist for the Certification of Trusted Digital Repositories" [1]. The checklist defines management policies that are organized into criteria for: The Organization; Repository Functions, Processes, and Procedures; The Designated Community & the Usability of Information; and Technologies & Technical Infrastructure. Each set of assessment criteria can be characterized as a set of rules applied to state information that define a specific management policy. The result of applying the rules can also be kept as state information within the trusted digital repository. An expression of the set of rules and state information is under development for preservation repositories built on the integration of DSpace [2] and the Storage Resource Broker (SRB) [3], and is available on request.

The rules assume that mechanisms exist within the preservation environment to ensure the integrity of both data and metadata. An essential component of a trusted digital repository is the ability to mitigate risk of loss of records and state information. We examine the types of mechanisms available within the Storage Resource Broker to assure the integrity of the digital repository. These mechanisms are available at all SRB installations, including SRB data grids installed outside of the San Diego Supercomputer Center.

1.1 RLG Assessment Criteria

The assessment criteria specify the policies that are needed for governance, sustainability, robustness, and use of the preservation facility. These policies in turn can be expressed as the set of expectations for assured data access and sustained linkage of preservation metadata to records. The RLG assessment criteria can be interpreted as categories of risk that must be managed for a preservation environment to be trustworthy.

1.2 Types of Risk

Managing data reliability requires protection against many types of risk, including:

Technology failure
- Storage media failure
- Vendor product errors
- Natural disaster
- Malicious user security breaches

Technology evolution
- Media obsolescence
- Format obsolescence
- Storage obsolescence
- Access mechanism obsolescence

Collection evolution
- Provenance metadata changes
- Name space degradation
- Semantics evolution (obsolescence of terms)

Organizational failure
- Storage operational errors
- Mismanagement of versions
- Loss of funding

The risks may be dynamic, requiring the ability to respond to faults that occur during data ingestion, data storage, and data retrieval. The faults may lead to data corruption (bad bits in the records), metadata loss, and data loss. A single copy of data or metadata is not sufficient to mitigate the risk of data loss. The use of a single facility also cannot mitigate against the risk of eventual data loss. Thus a viable preservation environment supports multiple copies of data and metadata, and provides mechanisms to synchronize the copies. By

choosing to replicate data across different types of storage media, across different vendor storage products, between geographically remote sites, and into deep archives, the technology failures can be addressed.

By supporting data virtualization, the ability to manage collection properties independently of the choice of storage system, most obsolescence issues can be addressed. Data virtualization enables the incorporation of new technology including new access protocols, new media, and new storage systems. Format obsolescence is managed through use of versions. Data virtualization also supports the evolution of the metadata schema, ensuring that new preservation authenticity attributes can be used over time. By choosing to federate independent preservation environments that use different sustainability models and different operational models, it is possible to address the organizational challenges.

2. Risk Mitigation Mechanisms

The Storage Resource Broker supports a wide variety of mechanisms that can be used to address the above risks. The mechanisms can be roughly divided into the following categories:

- Checksum validation
- Synchronization
- Replication, backups, and versions
- Federation

We examine the actual SRB Scommand utilities to illustrate how each of these types of integrity assurance mechanisms is used in support of the NARA Research Prototype Persistent Archive [4] and the NSF National Science Digital Library persistent archive [5].

2.1 Checksums

Checksums are used to validate the transfer of files, as well as the integrity of stored data. The SRB uses multiple types of checks:

- TCP/IP data transport checksum
- Simple check of file size
- Unix checksum (System5 sum command. Note that it does not detect blocks that are out of order)
- MD5 checksum (Robust checksum, implemented as a remote procedure)

The SRB uses TCP/IP to do all data transport. This implies that checksums are validated by the transfer protocol during all data movement. However the TCP/IP checksum is susceptible to multiple bit errors and may not detect corruption of large files. Also, the transport protocol does not guarantee end-to-end data integrity, from the submitting application to the final disk storage.

The preferred method is to checksum a file before transport, register the value of the checksum in the MCAT metadata catalog, and then verify the checksum after the transfer.

Sput –k *localfilename SRBfilename*
: Put a file into a SRB collection. Client computes simple checksum (System5 sum command) of the local file and registers with MCAT. No verification is done on the server side.

Sput –K *localfilename SRBfilename*
: Put a file into a SRB collection. After the transfer, the server computes the checksum by reading back the file that was just stored. This value is then compared with the source checksum value provided by the client. This verified checksum value is then registered with MCAT.

Sget –k *SRBfilename localfilename*
: Get a file from a SRB collection. Retrieves simple checksum (System5 sum command result) from the MCAT and compares with the checksum of the local file just downloaded.

The SRB provides mechanisms to list size and checksums of files that were loaded into a SRB collection. The utilities identify discrepancies between the preservation state information and the files in the storage repositories.

Sls -V
: Verifies file size in a SRB vault with file size registered in MCAT. Lists error when file does not exist or the sizes do not match.

Schksum -l *SRBfilename*
: Lists the checksum values stored in MCAT including checksums of all replicas.

Schksum *SRBfilename*
: If a checksum exists, nothing is done. If the checksum does not exist, it is computed and stored in MCAT.

Schksum –f *SRBfilename*
: Force the computation and registration of a checksum in MCAT.

Schksum –c *SRBfilename*
: Computes checksum and verifies value with the checksum registered in MCAT

Spcommand –d <*SRBfile*> "*command command-input*"
: Discovers where *SRBfile* is stored, and executes *command* at the remote site. The MD5 checksum is implemented as a proxy command.

2.2 Synchronization

The SRB supports the synchronization of files from system buffers to storage, between SRB

collections, and between a SRB collection and a local directory. The synchronization commands are used to overcome errors such as loss of a file, or a system halt during transfer of a file. In addition, the SRB server now incorporates mechanisms to validate the integrity of recently transferred files.

srbFileChk server

The srbFileChk server checks the integrity of newly uploaded files. By default, it runs on the MCAT enabled host. It can also be configured to run on other servers by changing the fileChkOnMes and fileChkOnThisServer parameters in the runsrb script.

A fairly nasty problem with a loss of data integrity can exist in recently uploaded files if the UNIX OS of the resource server crashes during the transfer. Data that are uploaded successfully may still be in a system buffer and not on disk. This problem is particularly bad because the SRB system has already registered the files in the MCAT and is not aware of the integrity problem until someone tries to retrieve the data. A typical symptom of this mode of corruption is the size of the corrupted file is not the same as the one registered in MCAT. The srbFileChk server performs a file check operation for newly created SRB and lists errors. By default, it wakes up once per day to perform the checking.

SphyMove –S *targetResource SRBfilename*

Sysadmin command to move user's files between storage resources. Since the original copy is removed after the transfer, the UNIX fsync call is executed after each successful SphyMove to ensure the files do not remain in a system buffer.

Srsync –s *Localfilename s:SRBfilename*

Synchronize files from local repository to SRB collection checking for differences in file size.

Srsync –s s:*SRBfilename Localfilename*

Synchronize files from SRB collection to local repository checking for differences in file size.

Srsync –s s:*SRBsourcefile s:SRBtargetfile*

Synchronize files between two SRB collections checking for differences in size.

Srsync *Localsourcefile s:SRBtargetfile*

Synchronize files from local repository to SRB collection using checksum registered in MCAT.

Srsync –l s:*SRBsourceCollection s:SRBtargetCollection*

List all the files that have different checksums between the source and destination SRB collections.

Synchronization commands are also used to ensure that the multiple copies of a file have the same bits.

Ssyncd *SRBfilename*

Synchronize all replicas of a file to be the same as the "dirty" or changed version.

Ssyncd –a –S *SRBresourcename SRBfilename*

Synchronize all replicas of a file, ensuring that a copy exists on each of the physical storage systems represented by the logical storage resource name *SRBresourcenam.*

Ssyncont *SRBcontainer*

Synchronize the permanent copy of a SRB container (residing on tape) with the cached copy residing on disk.

Ssyncont –z *mcatZone SRBcontainer*

Synchronize the SRB container residing in the data grid Zone *mcatZone* with the cached copy residing on disk in data grid Zone *mcatZone*

Synchronization between the metadata catalog and the storage repository is also needed, either to delete state information for which no file exists, or to delete files for which no state information exists.

Schksum -s *SRBfilename*

Does a quick check on data integrity using file size. Any discrepancies are listed as errors, including files not present on the storage system. This identifies bad state information.

vaultchk utility

This is a sysadmin tool that identifies and deletes orphan files in the UNIX vaults. Orphan files (files in SRB vaults with no entry in the MCAT) can exist in the SRB vault when the SRB server stops due to system crashes or server shutdown at certain critical points (such as during a bulk upload operation).

2.3 Replica, backups, and versions

The synchronization commands rely on the existence of multiple copies of the files. We differentiate between:

- Replica, a copy of a file that is stored under the same logical SRB file name.
- Backup, a copy of a file that has a time stamp.
- Version, a copy of a file that has an associated version number.

Replicas can be synchronized to ensure that they are bit for bit identical. A Backup enables the storage of a snapshot of a file at a particular time. Versions enable management of changes

to a file. Commands are needed to create each of these types of copies and to verify that identical copies are indeed identical.

Sreplicate *SRBfilename*
: Create a copy of a SRB file and register as a replica of the existing SRB file. Each time the command is executed another replica is made.

Sreplicate –l *localfilename SRBfilename*
: Load a local file into a SRB collection and register as a replica of an existing SRB file called *SRBfilename.*

Sreplcont –S *SRBresource SRBcontainer*
: Replicate a SRB container onto the named SRB resource.

Sbkupsrb –S *SRBresource SRBcollection*
: Backup a SRB collection to the SRB logical resource. First check whether a copy of the data already exists and that the copy is up to date. If not, create a copy on the resource.

Sbkupsrb –K–S *SRBresource SRBcollection*
: Verify the backup copy is indeed copied correctly by creating a checksum on the original, and then computing a checksum after the copy is made and comparing with the original.

Sput –n *replicanumber Localfilename SRBfilename*
: Load a local file into a SRB collection as a specific replica number or version.

SmodD –V *replicanumber oldVersionString newVersionString SRBfilename*
: Set a version identifier for the SRB file copy identified by *replicanumber.*

2.4 Federation

The SRB supports synchronization between two independent data grids. This provides a way to ensure that a copy of files and their metadata exist in a geographically remote data grid that can be under separate administrative control, using different storage resources, using different storage technology, and governed by a different sustainability model. The ability to create a copy of a collection ensures that the choice of storage technology or sustainability model does not constitute a risk.

Szonesync.pl –u –z *SRBZone*
: Sysadmin command to synchronize user information between the local data grid and the specified remote data grid *SRBZone.*

Szonesync.pl –d –z *SRBZone*
: Sysadmin command to synchronize data information between data grids. This is equivalent to the registration of files from the local data grid into the remote data grid.

Scp –S *Zoneresourcename SRBsourcecollection SRBtargetcollection*
: Copy each SRB file in the local zone source collection to the remote zone target collection and store files on the specified remote zone logical resource. Each collection name is written as: /zone/home/user.domain/collection Thus copying between data grids just requires specifying the zone name as part of the collection name.

Srsync s:*SourceCollection s:TargetCollection* /localzone/home/user.domain/sourcecollection /remotezone/home/user.domain/targetcollection
: Synchronize the files in SourceCollection with the files in TargetCollection using the checksums registered in the MCAT catalogs to detect differences.

3. Summary

A surprisingly large number of commands are required to implement the replication, synchronization, and federation mechanisms that are needed to meet assessment criteria for Trusted Digital Repositories. These mechanisms have been implemented in the Storage Resource Broker.

Acknowledgements

This project was supported by the National Archives and Records Administration under NSF cooperative agreement 0523307 through a supplement to SCI 0438741, “Cyberinfrastructure; From Vision to Reality”. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, the National Archives and Records Administration, or the U.S. government.

References

1. *An Audit Checklist for the Certification of Trusted Digital Repositories*. RLG, Mountain View,CA, August 2005. http://www.rlg.org/en/pdfs/rlgnara-repositorieschecklist.pdf
2. DSpace, http://www.dspace.org/
3. Storage Resource Broker, http://www.sdsc.edu/srb/
4. NARA Research Prototype Persistent Archive, http://www.sdsc.edu/NARA
5. National Science Digital Library, http://nsdl.org/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

APPLICATION OF THE NERC DATA GRID METADATA AND DATA MODELS IN THE NERC ECOLOGICAL DATA GRID

Neil Bennett[1], Rod Scott[2], Mike Brown[2], Kevin O'Neill[3], Mandy Lane[2], Andrew Woolf[3], Kerstin Kleese-van Dam[1], John Watkins[2]

[1]CCLRC – Daresbury Laboratory, Daresbury, Warrington, Cheshire, WA4 4AD, UK.
[2]CEH - Lancaster Environment Centre, Library Avenue, Bailrigg, Lancaster, LA1 4AP, UK.
[3]CCLRC – Rutherford Appleton Laboratory, Chilton, Didcot, Oxon, OX11 0QX, UK.

Abstract: The Centre for Ecology & Hydrology (CEH) holds various databases collectively representing a valuable environmental research resource. However, their use inside and outside CEH is constrained by lack of data accessibility and interoperability. This project has focused on three test-bed datasets held at the Lancaster Environment Centre which form a good example of the diversity of CEH terrestrial and freshwater data. Metadata systems and access tools have been constructed in collaboration with the NERC Data Grid (NDG) to provide users with Grid services linking data discovery to dataset delivery. This paper focuses on the modelling of CEH's data and metadata using the NDG models.

1. Introduction

CEH is the leading UK body for research, survey and monitoring in terrestrial and freshwater environments. The main aim of this project is to demonstrate that a diverse subset of CEH's data can be made more accessible by integration into the NDG. Three datasets have been used for the EcoGrid project: the Countryside Survey (CS) vegetation database [7]; the Environmental Change Network (ECN) database [8]; and, the Lakes database [9].

NDG has defined a detailed metadata model [1], MOLES (Metadata Objects for Links in Environmental Science) and a data model [2] known as CSML (Climate Science Modelling Language). Both models are standards-based and represented as XML schemas. They are intentionally generic so they can accommodate data from a wide range of scientific disciplines.

MOLES has been designed to allow production of the various 'industry standard' discovery formats, such as DIF [10], ISO 19115 [11], FGDC/GEO [12] and SensorML [13]. MOLES allows a smooth link from data browse to data usage.

CSML provides information about the data that processing and visualisation services need. It describes the data in semantic terms and virtualises it, removing the need to know the actual format of the data.

To integrate the EcoGrid data into NDG requires mapping to the NDG models [6]. Often the distinction between data and metadata is blurred. Theoretically, metadata are descriptive data whereas data are physical measurements. However, in reality, measurements can be part of a descriptive coding system, e.g. to search datasets by species of interest requires the individual species codes recorded in each data set to be in the discovery metadata. Thus, species will feature in both MOLES records (from which discovery metadata is generated) and CSML.

For discovery metadata such as species, it is important that EcoGrid recognises synonyms. Otherwise, a user searching the portal may not retrieve all relevant records. EcoGrid is collaborating with the producers of a data dictionary, using a common taxonomical classification, to solve this problem.

Also, a significant proportion of EcoGrid data relates to freshwater chemistry where various parameters are measured and values recorded in appropriate units. To help describe datasets fully and aid comparison with other datasets both inside CEH and outside, it is important to reference units and parameters to standardised definitions in widely accepted dictionaries. CSML incorporates references to unit and 'phenomenon' dictionaries so EcoGrid's use of CSML will help with the issues of accessibility and interoperability.

2. Securing Sensitive Data

Much of the testbed data is sensitive so it is important that users do not receive any more information than they need. NDG security takes care of user authentication and authorisation [5]. Its development used the CCLRC Data Portal Authorisation Architecture [3] as a starting point.

Where appropriate, CEH have implemented a two tier data structure so that authorised users can access raw data whereas summarised data is available to everyone. For both ECN and Lakes, the data has been summarised temporally.

For CS, the raw data is accessible but the precise location of the survey is not disclosed. Thus, datasets are available at the CS square level but the location shown is just the government office region. Hence, only one authorisation role is required.

3. The Elements of MOLES

To gain an understanding of how the test data has been modelled and the issues involved, it is important to introduce the models themselves.

There are four main elements within MOLES:

- Activity, e.g. a whole project or a field survey in a particular location.
- Data Production Tool. Used to make measurements and collect data, e.g. something as simple as a quadrat.
- Observation Station. A site of data production tools , e.g. a nature reserve.
- Data Entity. This contains metadata about the data itself and is linked to usage metadata (in a CSML file).

These elements are linked by IDs to create deployments (the use of a data production tool at an observation station to produce a data entity on behalf of an activity). Much of CEH's data is collected by scientists undertaking surveys in the field. MOLES handles this by classifying the surveyor as an observer within the deployment.

4. Introduction to CSML

There are several components within CSML.

Phenomenon. This is a property or variable that an activity sets out to measure. This can be referenced to a standard dictionary (such as Climate Forecast standard names [14])

Unit. The unit of measure of a phenomenon. Units can be referenced to a standard dictionary such as Unidata's 'udunits' [15].

Feature Type. An abstract representation of the measurement of a phenomenon, e.g. a measurement taken at one point over a time series is a PointSeriesFeature.

Array. Sometimes, a temporal or spatial coordinate series is used repeatedly. It is then more efficient to declare an Array once and make references to it than explicitly write out the series every time.

Reference System. These provide efficiencies when there is a systematic pattern in (say) spatial coordinates or a time series. This means that every time or spatial coordinate need not be specified explicitly. Instead, we can specify a starting point, an interval and an end-point or total number of elements.

5. Modelling data using CSML

5.1 Water chemistry & meteorological data

CSML was developed by the Atmospheric Science and Oceanography communities and as such copes very well with water chemistry & meteorological data. Such data is characterised by chemical and physical measurements taken at single points in space over time series.

5.2 Species

Each CEH dataset may contain data for hundreds of different species. These species names could be rolled up into phenomenon names e.g. 'count of bat species A', 'count of bat species B', etc. However, this would require huge numbers of different phenomena within a dataset that can not be matched to standard phenomenon dictionaries. The solution implemented is to create an EcoGrid dataset for each species.

5.3 Categorical Data

CEH has a significant amount of categorical and free-text type data (e.g. habitats), which should clearly be represented as phenomena. Unlike most phenomena, these do not possess units and so will not possess a 'unit' attribute.

5.4 Missing values

A common scenario is that measurements are made over a time series. However, for CEH, occasionally there are breaks in the series, perhaps due to equipment being shut down for maintenance, etc. Potentially, this could result in more times in the series than there are measurements recorded. CSML copes well with this by allowing us to explicitly state when a "missing value" occurs.

6. Issues & Potential Solutions

6.1 Species

One of the main limitations of CSML is that it has no obvious way of modelling biological concepts that make up much of CEH's data. Often the number of individuals of a species is counted and this can be subdivided by gender, morph, stage of development, etc in places. There are two possible solutions.

Solution 1

The various combinations of these attributes could be rolled up into phenomenon names. This is conceptually simple but creates a few problems. Firstly, a very large number of phenomena would be required. Secondly, these phenomena would be so fine grained that they can not be referenced to a standard dictionary, and other researchers would find it difficult to compare their results with CEH's. Finally, the phenomenon name used in the MOLES file would be a simple name (such as number of individuals) and as such would not match up to the phenomenon name in the corresponding CSML file.

Solution 2

Another solution is the use of composite phenomena [4], a concept inherited by CSML. This consists of another phenomenon (e.g. count of individuals of species A) to which a vector is applied. In this example, the vector could be gender (e.g. male and female). It is also possible to nest composite phenomena, so that a composite phenomenon consists of another composite phenomenon (and a vector) which in turn consists of a simple phenomenon (and another vector). So, extending the example, could result in:

Composite phenomenon 1 (count of individuals by species by gender) = composite phenomenon 2 (count of individuals by species) * vector 1 (gender)

AND

Composite phenomenon 2 = Simple phenomenon (count of individuals) * vector 2 (species)

An advantage of this approach is that the root phenomenon is simple and standardised. However, a major problem is that the 'Composite Phenomenon' construct of OGC is under development [4], and CSML's associated tools do not currently support it. An additional problem is that it is not clear how the relationship between data values and vectors would be illustrated. For example, if a phenomenon applies across a range of locations then the string of values measured will be of the form "x1 x2 x3". If the phenomenon is complex and contains a vector (say gender) applied to a simple phenomenon then the string could be of the form "(x1 y1), (x2 y2), (x3 y3)". It is not then clear whether y1 is the phenomenon witnessed for gender 1 at location 2 or whether it is for gender 2 at location 1. The problem gets worse as the level of nesting increases.

For now, this issue has been circumnavigated by making data available only at the 'number of individuals' level. In the future, ontologies may be used to cope with such situations better.

6.2 Profile Series Features

CSML has a feature type called ProfileSeriesFeature to represent a measurement recorded at various points along a directed line in space over a time series.

CEH has lake data where measurements are taken at various water depths over a time series. Here, the ProfileSeriesFeature type would be ideal for modelling such data. However, since the measurement depths may not be constant from one time to the next, ProfileSeriesFeatures can not be used. This situation is encountered in other domains (e.g. observational oceanography) and a new feature type will be introduced to cope with such eventualities in a future revision of CSML.

6.3 Observations & Measurements paper

CSML inherits part of its structure from the schema detailed in the Open Geospatial Consortium Observations & Measurements paper [4]. This paper contains an example of how this model could be applied to an ecological survey example, and has been reviewed to see whether it provides solutions to the issues faced when using CSML. However, the data modelled in the example is far simpler than what CEH has so it has not provided a complete solution.

6.4 Transects

Some of CEH's data (e.g. butterfly surveys) is collected via transects. Transects are just routes

that the observer walks along. As the observer follows the transect, they make observation notes at various points. CSML has a Trajectory feature type that might be used to model transect data. However, data is repeatedly collected from a transect forming a time series. Unfortunately, trajectories can not be used to model time series.

An alternative is the ProfileSeriesFeature type. However, a profile must be a straight line with a specified direction, whereas a transect may be a curved path. This problem is avoided by summing data at a higher level (e.g. site).

6.5 Sublocations

For some of its data, CEH uses several different subdivisions to express location more precisely, e.g. for spittle bugs there are sites comprising locations that comprise quadrats. This situation is not handled by CSML well. One option is to use reference systems but this would be very complex to implement and understand. Again, the solution adopted has been to sum data at a higher level (e.g. site or location code).

6.6 Date type phenomena

In some cases, there is date-type data that does not reflect the time at which measurements were made but instead is one purpose of the survey itself. For example, one survey measures when frogs are first seen congregating, hatching and leaving the pond. This data must therefore be modelled as phenomena. It is possible to do this if we do not specify a related unit. However, this implies that the date is just a text string when in fact it carries more meaning than that.

If CSML allowed us to specify the corresponding unit as 'date' then a user would be able to make sensible comparisons between data e.g. to see where frogs are hatching first.

6.7 Date Type Parameters

Typically, a measurement is made at a point in space over a time series. However, there are cases, particularly for water chemistry, where samples are taken from a river and then at some subsequent time/date, analysis is done. There can be several stages for the analysis and the time/date of each is recorded, e.g. pH measurement, filtration, and completion of analysis.

There is currently no suitable place for these within CSML. They are not phenomena because they are not something a scientist sets out to measure. On the other hand, it is possible to record one time/date as an input parameter but not several. Such data is omitted from CSML files.

6.8 Replicate measurements

For water chemistry measurements, CEH sometimes uses multiple test tubes at the same location at the same time to measure the same phenomena. There is no mechanism within CSML for modelling this.

Composite phenomena could be used with a vector containing the replicate IDs. However, these are not currently supported in CSML. At this stage the only possible solution is to store mean or modal values of these replicates.

6.9 Protocol metadata

To study freshwater invertebrates, CEH uses nets to collect samples from rivers and lakes. The properties of a net are significant in determining which species it yields. Technically, the net is a data production tool and as such it would normally be described in the corresponding MOLES record. However, several different nets can be used within one dataset or study so this information must go into the CSML record instead. However, there is currently no suitable place for such information within CSML.

7. Future

It would be desirable to extend EcoGrid to cover all CEH data and also incorporate data held by the National Biodiversity Network which focuses on Sites of Special Scientific Interest and thus is complementary to EcoGrid's data.

UK ecologists would like to collaborate with ecologists worldwide. The Knowledge Network for Biocomplexity [16] is trying to solve similar problems and has developed Ecological Metadata Language (EML) to describe ecological data. Creation of EML records should open up CEH's data to a much wider audience.

Environmental data always has a spatial component. The first version of NDG will allow spatial searching of datasets. However, nothing more complex is permitted. The second stage of NDG, which began in late 2005, may introduce improved spatial capabilities with consequent benefits for EcoGrid.

8. References

[1] A specialised metadata approach to discovery and use of data in the NERC Data Grid. K O'Neill et al., Proceedings of the UK e-Science All Hands Meeting 2004. (http://www.allhands.org.uk/2004/proceedings)

[2] Climate Science Modelling Language: Standards -based markup for metocean data", 85th meeting of American Meteorological Society, San Diego, Jan 2005.

[3] Grid Authorisation Framework for the CCLRC Data Portal. A Manandhar et al., Proceedings of the UK e-Science All Hands Meeting 2003.

[4] Observations and Measurements, OGC Discussion Paper 05-087r3, Simon Cox editor. http://portal.opengeospatial.org/files/?artifact_id=14034

[5] NERC Data Grid Authorisation Architecture. N Bennett et al., Proceedings of the UK e-Science All Hands Meeting 2005. (http://www.allhands.org.uk/2005/proceedings)

[6] NERC Ecological Data Grid. N Bennett et al., Proceedings of the UK e-Science All Hands Meeting 2005. (http://www.allhands.org.uk/2005/proceedings)

[7] CEH Countryside Survey. http://www.cs2000.org.uk

[8] CEH Environmental Change Network. http://www.ecn.ac.uk

[9] CEH Lakes Database. http://www.ceh.ac.uk/sections/eaf/EAFCumbriaLakesDatabase.html

[10] Directory Interchange Format. http://gcmd.nasa.gov/User/difguide/difman.html

[11] ISO/TC211. ISO activity in the Geographic Information/Geomatics domain including the ISO191xx series of standards. http://www.isotc211.org/

[12] FGDC. Federal Geographic Data Committee Standard for Digital Geospatial Metadata (FGDC-STD-001-1998). http://www.fgdc.gov/metadata/metadata.html

[13] Open Geospatial Consortium – SensorML. http://www.opengeospatial.org/

[14] CF standard name table. http://www.cgd.ucar.edu/cms/eaton/cf-metadata/standard_name.html

[15] udUnits. http://www.unidata.ucar.edu/software/udunits/

[16] http://knb.ecoinformatics.org/software/eml/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

The GridSite Proxy Delegation Service

Andrew McNab, Shiv Kaushal

Department of Physics and Astronomy, University of Manchester

Abstract

X.509 Proxy Certificates may now be delegated from a client to a service using the GridSite/gLite Delegation Web Service. We have implemented both a client and an implementation of the service, and we describe the delegation process in terms of the operations of the delegation portType. We also explain how proxies delegated to a service can be used by other Web Services written as CGI scripts or executables, and how other components of GridSite and Unix account permissions can be combined to control which services can access a delegated proxy credential on a shared server. This work has been done as part of the GridSite and EGEE projects, and the GridSite toolkit now includes functions which third-party developers can use to add support for a delegation portType to their own applications.

1. Introduction

X.509 Proxy Certificates[1] were originally introduced by the Globus Project[2] and have subsequently become central to the operation of international production Grids, such as the LHC Computing Grid[3] and the EGEE[4] project. A proxy certificate allows jobs or agents to prove that they are running on behalf of a particular user, and they are granted to jobs by some form of delegation procedure involving the creation and signing of a short-lived X.509 certificate. This paper describes an X.509 Proxy Certificate delegation web service developed jointly as part of the EGEE and GridSite[5] projects.

2. The delegation protocol

2.1 X.509 delegation

RFC 3280[1] describes how the certificate requests and proxy certificates (PC) are processed by the service and client. The GridSite delegation service caters to the scenario of a remote web service which requires a PC, and a client which already possesses either a full X.509 end entity certificate (EEC) issued by a Certification Authority, or an X.509 PC previously created by the user.

The delegation procedure involves the following steps:

- The service is contacted by the client, which asks for an X.509 certificate request.
- The service then generates an RSA public and private key pair, stores the private key and uses the public key as the basis of an X.509 certificate request.
- The service may choose to pre-populate the request with values, such as the desired expiration time and the uses to which the proxy may be put.
- The service returns the certificate request to the client, which enforces any restrictions it places on values such as expiration time, and then signs the request using the client's own private key.
- The resulting X.509 PC is then sent to the service, which now possesses the corresponding private key, the PC itself and all of the X.509 PC and EEC which prove the chain of trust back to a trusted Certification Authority.

Since an X.509 PC is very similar to a conventional EEC, it can be used within most client applications without modification. On the server side, for applications which use SSL/TLS[6] such as HTTPS[7], only the code which checks the client's certificate chain requires modification (to permit chains including PCs in addition to Certification Authority certificates and a final EEC.) For Apache/mod_ssl, this modification is done dynamically by GridSite by intercepting the underlying OpenSSL[8] callbacks during the authentication phase. PC support was also added to OpenSSL itself in version 0.9.7g.

2.2 Web Service operations

To allow clients and services to perform this two stage delegation procedure, a delegation web service specification has been developed within the EGEE Middleware Security Group[9] for use by the GridSite and gLite[10] middleware systems.

This paper describes version 1.1.0 of the service (http://www.gridsite.org/delegation-1.1.0.wsdl) which uses the namespace http://www.gridsite.org/namespaces/delegation-1.

The service consists of a single portType, Delegation, which may either be implemented as a standalone web service or as an additional portType of services which require delegation. In either mode, a Delegation ID string is used to identify the delegation session, which is necessary if the same client wishes to maintain multiple simultaneous delegations to the same server, with different expiration times or other options. This scenario is likely to occur if a user submits multiple jobs to a grid which are unaware of each other but arrive at the same site.

The delegation procedure is carried out using two operations within the portType, getNewProxyReq and putProxy.

To initiate a delegation, the client sends an empty getNewProxyReq request, and receives a Delegation ID generated by the service and an X.509 certificate request in Base64 encoded PEM format.

After populating and signing the request, the client then completes the procedure using the putProxy operation to send the signed X.509 PC back to the service, and includes the Delegation ID to allow the service to associate the PC with the private key generated in the first phase.

The renewProxyReq operation is also provided which can be used in place of getNewProxyReq and allows the client to specify an existing Delegation ID and extend the lifetime of its delegation session. In return, the service responds with an X.509 certificate request and putProxy is used as before.

Clients can enquire about the status of a delegation session with the getTerminationTime operation, which returns the expiration time of the proxy corresponding to the given Delegation ID; and the destroy operation can be used to remove a proxy including the private key from the service's store.

3. The GridSite implementation

As mentioned above, GridSite adds support to Apache[11] for the authentication of clients using an X.509 PC with an HTTPS connection. Web services can then be written as CGI scripts or executables, and obtain the authentication information, including the certificate chain used, from the CGI environment variables.

3.1 GridSite Delegation Service

The GridSite Delegation Service uses this environment and is written in C with the aid of standard toolkits. The gSOAP[12] web services toolkit is used to convert between SOAP messages and C structures, and OpenSSL is used to examine X.509 certificates and generate private keys and certificate requests. All the additional utility functions which were required, including the storage of proxies, were written as part of the GridSite library, and where they can be used by third-party authors to add a delegation portType to their own web services.

3.2 htproxyput client

htproxyput is a command-line client which can be used with the GridSite Delegation Service or other implementations of the portType. The command syntax is similar to the htcp family of commands supplied with the main GridSite distribution, and its default options look for the user's X.509 EEC or X.509 PC in commonly used locations, including the X509_USER_PROXY environment variable, and the "/tmp/x509up_uN" file created by Globus grid-proxy-init or gLite voms-proxy-init.

As with the Delegation Service, most of the functionality is provided by calls to GridSite toolkit functions, supplemented by gSOAP and OpenSSL.

4. Use with CGI services

The implementation of the delegation service has been designed to allow the sharing of proxies with other web services hosted on the same GridSite enabled server. GridSite allows Web Services for Grids to be written as CGI scripts or executables, with authentication and access control decisions performed within Apache by the mod_gridsite module, which also makes the values of X.509 PC and EEC available, plus any VOMS X.509 Attribute Certificates. GridSite also provides a setuid wrapper, based on Apache's suexec, which allows Unix accounts and file permissions to control what areas of the hosting server are accessible to each instance of a service.

4.1 Proxy Storage

The delegation service must store the private key when it is generated and the signed X.509 PC when it is returned by the client. Both filesystem and SQL database stores have been specified. GridSite implements file-based storage, which interoperates with GridSite's use of Unix file permissions to sandbox other services and sessions owned by different users.

All of the proxy files are stored in subdirectories of a proxycache directory, which is a sibling of the DocumentRoot directory specified in the Apache virtual server configuration.

For example "/var/www/proxycache" if "/var/www/html" is the DocumentRoot. This allows multiple virtual servers, with different DNS components in their URL, to be hosted on the same physical server, without clashes between Delegation IDs which are only unique to each DNS domain name.

Within the proxycache directory, the cache subdirectory contains private keys awaiting the upload of the corresponding X.509 PC, organised into directories named after the URL-encoded Distinguished Name of the X.509 EEC, with one subdirectory for each Delegation ID.

For example
"/var/www/proxycache/cache/%2FC%3DUK%
2FO%3DeScience%2FOU%3DManchester%
2FL%3DHEP%2FCN%3Dandrew%
20mcnab/acd5ab55a6c9473e/"

Once the signed X.509 PC is returned by the client, the PEM encoded proxy chain including the EEC and PC and private key is saved as userproxy.pem within a subdirectory named after the Delegation ID, which is itself a subdirectory of the user's URL-encoded EEC Distinguished Name. For example,
"/var/www/proxycache/%2FC%3DUK%2FO%
3DeScience%2FOU%3DManchester%2FL%
3DHEP%2FCN%3Dandrew%
20mcnab/acd5ab55a6c9473e/"

4.2 Interaction with gsexec

GridSite's gsexec wrapper allows CGI scripts or executables to run as users other than the apache user, which typically owns the listening Apache processes and most of their files. The server can be configured to assign a pool account to each client X.509 EEC identity, or to each directory containing one or more services, or a mixture of both on a per-directory basis. These modes allow the sandboxing of instances of a service associated with different clients, or of sets of services owned by different application groups.

Since the delegation service stores delegated proxy chain and private key in the Unix filesystem, ownership and permissions can be

used to control access to the proxy in a transparent way.

Let us assume that Apache runs as user apache and group apache; that each pool user has both a user and its own group; that the apache user has secondary membership of each pool user's private group; and that files are created with owner and group have read/write permission.

This arrangement lets Apache write to files, if it receives an authorized PUT request, and allows CGI scripts to run as a given pool user and write files, but prevents them from writing to files owned by other pool users.

If the GridSite Delegation Service is then run as apache, it has the ability to create proxy files which are only readable by the pool user which corresponds to the correct EEC identity, by virtue of its group readable permission and the pool user's private group.

5 Summary

We have implemented both a client and a service for the GridSite/gLite Web Service for the delegation of X.509 Proxy Certificates. This work has been done as part of the GridSite project, and the GridSite toolkit now includes functions which third-party developers can use to add support for a delegation portType to their own applications.

Acknowledgements

This work has been funded by the UK's Particle Physics and Astronomy Research Council, through its e-Science Studentship programme and support for the GridPP collaboration.

The GridSite/gLite delegation web service is the result of discussions within the EGEE Middleware Security Group, and we would like to thank Akos Frohner, Joni Hahkala, Olle Mulmo and Ricardo Rocha for their work on finalising the specification.

References

1. RFC 3820, “Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile”, S. Tuecke et al, June 2004.
2. The Globus Project, http://www.globus.org/
3. LHC Computing Grid, http://www.cern.ch/lcg
4. Enabling Grids for E-Science, http://www.eu-egee.org/
5. The GridSite Project, http://www.gridsite.org/
6. RFC 2246, “The TLS Protocol”, T. Dierks et al., January 1999.
7. RFC 2818, “HTTP over TLS”, E. Rescorla, May 2000.
8. The OpenSSL Project, http://www.openssl.org/
9. The EGEE Middleware Security Group, http://egee-jra3.web.cern.ch/egee-jra3/
10. gLite, http://glite.web.cern.ch/glite/
11. The Apache Webserver, http://httpd.apache.org/
12. gSOAP, http://gsoap2.sourceforge.net/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Survey of Major Tools and Technologies for Grid-enabled Portal Development

Xiaoyu Yang, Martin T. Dove, Mark Hayes, Mark Calleja, Ligang He, Peter Murray-Rust

University of Cambridge, Cambridge, UK

Abstract

Grid portals that can provide a uniform access to underlying grid services and resources are emerging. Currently, there are a variety of technologies and toolkit that can be employed for grid portal development. In order to provide a guideline for grid portal developers to choose suitable toolkit, in this paper, we briefly classify grid portals into non portlet-based and portlet-based, and attempt to survey major tools and technologies that can be employed to facilitate the creation of grid portals.

1 Introduction

Grid portals can provide an integrated platform for end users to access grid services and resources via Web browsers without the need to download and install specialised software packages and libraries [2]. Currently, there are a variety of development tools, frameworks and components that can support the grid portal development. However, the features and usage of these tools, frameworks and components can be different. In order to provide a guideline for grid portal developers to choose appropriate toolkit, this paper aims to survey major grid portal development tools and technologies that can be employed to facilitate grid portal development.

2 Grid Portals: Non Portlet-based Vs. Portlet-based

According to the way of building portals, we can briefly classify grid portals into non portlet-based and portlet based.

Non-portlet based - Many early Grid portals or early version of existing Grid portals are non-portlet based, for example, Astrophysics Simulation Collaboratory (ASC) portal [3], UNICORE [4,5], etc. These grid portals provide a uniform access to the grid resources. Usually these portals were built based on typical 3-tier web architecture: (i) Web browser, (ii) application server/Web server which can handle HTTP request from the client browser, and (iii) backend resources that include computing resources, databases, etc.

Portlet-based portal - A portlet is a Web component that generates fragments – pieces of markup (e.g. HTML, XML) adhering to certain specifications. Fragments are aggregated to form a complete web page. Developing portlet-based portals can bring many benefits to both end-users and developers, which now gets more recognition [6]. This can be reflected through evolution of some grid portal projects. For example, although ASC portal [3] did provide functionalities for astrophysics community to remotely compile and execute applications, it was difficult to maintain when the underlying supporting infrastructure evolved. Eventually the ASC portal was retired and its functionality moved into the Cactus portal developed by adopting GridSphere [8]. Another example is the GridPort portal [9]. The early GridPort was implemented in Perl and made use of HotPage [10] technology for providing access to grid access. Now the GridPort 4.0.1 adopts GridSphere. Similarly, the upcoming releases of Java CoG kit 4 will support the portlet-based development as well [11].

Two portlet standards, i.e. JSR-168 [12] and WSRP [13] (Web Services for Remote Portlets) can ensure portlets pluggable and independent on actual portal frameworks.

3 Survey of Major Portal Tools and Technologies

3.1 Commodity Grid (CoG)Kits

CoG toolkits [11] provide native and wrapper implementations of Globus [14]; for example, it provides the implementation of GSI, gridFTP, myProxy, and GRAM client implementations. It uses and leverages existing commodity frameworks, technologies, and toolkits in

cooperation with Grid technologies. CoG is often used to build Grid portals [11]. It includes Perl CoG, Python CoG and Java CoG.

3.2 GPDK

GPDK was a widely used toolkit for building non-portlet based portals. GPDK itself was built on top of the Java CoG based on standard 3-tier architecture [2]. GPDK was a successful product in creating application specific portals by many research groups such as the GridGaussian computational chemistry portal, the MIMD Lattice Computation (MILC) portal, etc [2]. However, GPDK tightly coupled presentation with logic in its design and did not conform to any portal/portlet specifications. This can result in poor extensibility and scalability. Now the GPDK is not supported any longer, and the major author of GPDK has moved to GridSphere.

3.3 GridSphere

The development of GridSphere has combined the lessons learned in the development of ASC portal and GPDK. The GridSphere Portal Framework [8] is developed as a key part of the European project GridLab [15]. It provides an open-source portlet-based Web portal, and can enable developers to quickly develop and package third-party portlet web applications that can be run and administered within the GridSphere portlet container. One of the key elements in GridSphere is that it supports administrators and individual users to dynamically configure the content based on their requirements [6]. Another distinct feature is that GridSphere itself provides grid-specific portlets and APIs for grid-enabled protal development. The main disadvantage of the current version of GridSphere (i.e. GridSphere 2.1) is that it does not support WSRP.

3.4 GridPort

The GridPort Toolkit [9] enables the rapid development of highly functional grid portals. It comprises a set of portlet interfaces and services that provide access to a wide range of backend grid and information services. GridPort 4.0.1 was developed by employing GridSphere. GridPort 4.0.1 might be the last release as the GridPort team has recently decided to shift the focus from developing a portal toolkit to developing production portals [9].

3.5 LifeRay Portal

The Liferay portal [16] is more than just a portal container [17]. It comes with helpful features such as Content Management System (CMS), WSRP, Single Sign On (SSO). It is open-source, 100 % JSR portlet API and WSRP compliant. Liferay is suitable for enterprise portal development. Institutions and companies that adopted Liferay to create their portals include *EducaMadrid, Goodwill, Jason's Deli, Oakwood, Walden Media*, etc [16].

3.6 eXo platform

The eXo platform [18] can be regarded as a portal and CMS [17]. The eXo platform 1 was more like a portal framework. The eXo platform 2 proposed a *Product Line Strategy* [19] as it is realised that end user customers need to get ready to use packaged solutions instead of monolithic product. The eXo platform 2 is now a core part on which an extensive product line can be built [19]. It features that it adopts Java Server Faces and the released Java Content Repository (JCR – JSR 170) specification. The eXo platform 2 adopts JSR-168 and supports WSRP.

3.7 Stringbeans

Stringbeans [20] is a platform for building enterprise information portals. The platform is composed of three components: (i) a portal container/server, (ii) a Web Services platform, and (iii) a process automation engine. At this time the portal server and Web services platform have been released. The process automation engine will be released in the near future [20]. It is JSR-168 compliant and supports WSRP. The Stringbeans was used for the UK National Grid Service (NGS) portal [21].

3.8 uPortal

uPortal[22] is a framework for producing campus portal. It is now being developed by the JA-SIG (Java Special Interest Group). It is JSR-168 compliant and supports WSRP. uPortal is now widely used in creating university portals, for example, the Bristol university staff portal.

3.9 OGCE (Open Grid Computing Environment)

The OGCE (Open Grid Computing Environment) project was established to foster collaborations and sharable components with portal developers [23]. OGCE consists of a core set of grid portlets that are JSR 168 compatible. Currently OGCE 2 supports GridSphere and uPortal portlet containers.

	JSR-168 compliant	WSRP compliant	Grid-specific portlets	Open source	Notes
Java CoG 1.2	✗	✗	-	✓	Development of grid portal framework, grid service, etc
GPDK	✗	✗	-	✓	Not supported any longer
GridSphere 2.1.4	✓	✗	✓	✓	**UK**: RealityGrid [29], MyGrid [30] GeneGrid [31], p-GRADE portal [32], SAKAI VRE Portal Demonstrator [33], etc **US**: Cactus [6], etc **EU**: GridLab, etc
GridPort 4.0.1	✓	✗	✓	✓	Developed based on GridSphere. Last release
Liferay 3.6.1	✓	✓	✗	✓	Widely used in developing portals
eXo 2	✓	✓	✗	✓	Widely used in many company portals
Stringbeans 3.0.1	✓	✓	✗	Dual licenses	Used in UK National Grid Service portal, etc
uPortal 2.5.1	✓	✓	✗	✓	Widely used in many universities
OGCE 2	✓	✗	✓	✓	Support GridSphere and uPortal
Pluto	✓	✗	✗	✓	Simple portlet container
Jetspeed -2	✓	✓	✗	✓	Widely used in creating portals
IBM WebSphere Portal 6.0	✓	✓	✗	Free for research	Widely used in creating portal at enterprise level. Free for research under *the IBM Academic Initiative.*

Table 1 Grid portal development toolkits comparison matrix

3.10 Pluto

Pluto is a subproject of Apache Portal project. It is the reference implementation of the JSR 168 [24]. Pluto simply provides a portlet container for portal developers to test the portlets, and does not provide many specific portlets.

3.11 Jetspeed

Jetspeed is another Aapche Portal project, which includes Jetspeed-1 and Jetspeed-2. Jetspeed-1 "provides an open source implementation of an Enterprise Information Portal, using Java and XML" [25]. Jetspeed-2 is the next-generation enterprise portal, and offers several architectural enhancements and improvements over Jetspeed-1 [26]. Jetspeed is more sophisticated than Pluto. Jetspeed is concentrated on portal itself rather than just a portlet container.

3.12 IBM WebSphere Portal

IBM's WebSphere Portal [27] is a framework that includes a runtime server, services, tools, and many other features that can help integrate an enterprise into a single, customizable interface portal. It implements the JSR 168 Portlet API and WSRP [27]. WebSphere Portal is a powerful tool and is widely used in many business companies and enterprises. IBM also provides a scheme called *the IBM Academic Initiative* [28] for academic and research institutes. Membership in *the IBM Academic Initiative* can have the latest technology and majority IBM software to use for free, which includes IBM WebSphere Portal.

4 Comparison and Findings

Having surveyed the major grid portal development tools, frameworks, and components, a comparison matrix table is produced as shown in Table 1. The evaluation criteria include (i) JSR-168 compliant, (ii) WSRP compliant, (iii) provision of grid-specific portlets and (iv) open source.

The table has revealed that:

1) Developing portlet-based grid portals now gets more recognition.

2) GridSphere has been widely used for grid-enabled portal development. GridSphere itself provides grid-specific portlets. The main disadvantage of the current version is that it dose not support WSRP.

3) Other open source portal frameworks and portlet containers such as uPortal, liferay, eXo, Jetspeed-2, etc are also appropriate for grid-enabled portal development. Although they do not directly provide grid-specific portlets, the existing open source JSR-168 grid portlets (e.g. OGCE, GridPort) can be reused or new grid portlets need to be developed.

4) For developing commercial grid portals, the toolkit of IBM WebSphere portal would be a good choice. Likewise, although WebSphere portal itself does not provide grid-specific portets, we can reuse the existing JSR-168 grid portlets or develop new ones. Under *the IBM Academic Initiative*, academic and institutional researchers can use the IBM WebSphere portal for free.

5) GridPort provides grid-enabled portlets, but GridPort 4.0.1 might be the last release as the team will now focus on the production portal development instead of portal toolkit.

Acknowledgements

Many thanks to DTI funded *MaterialsGrid* project.

References

[1] J. Novotny "Developing grid portlets using the GridSphere portal framework", *IBM developerworks,2004*

[2] J. Novotny "The Grid Portal Development Kit" *Concurrency and Computation: Practice and Experience, Special Issue: Grid Computing Environments*, vol. 14, no. 13-15

[3] G. Allen, G. Daues, I, Foster et al " The Astrophysics Simulation Collaboratory Portal: A science Portal Enabling Community Software Development" *Proceedings of the 10th IEEE International Symposium on High performance Distributed Computing* 2001

[4] UNICORE: http://unicore.sourceforge.net/

[5] M. Romberg "The UNICORE architecture: seamless access to distributed resources". Proceedings of the *8th IEEE International Symposium on High performance Distributed Computing* 1999

[6] I. Kelley, M. Russell, J. Novotny et al (2005) "The Cactus portal" *APAC'05*

[7] G. Allen, G. Daues, I, Foster et al " The Astrophysics Simulation Collaboratory Portal: A science Portal Enabling Community Software Development" *Proceedings of the 10th IEEE International Symposium on High performance Distributed Computing* 2001

[8] GridSphere: http://www.gridsphere.org

[9] GridPort: http://gridport.net/main/

[10] J. Boisseau, S. Mock, M. Thomas "Development of Web toolkits for computational science portals: The NPACI HotPage". *Porceeding of the 9th IEEE international Symposium on high performance distributed computing 2000*

[11] Cog Kit: http://www.globus.org/cog/java/

[12] Introduction to JSR 168 http://developers.sun.com/prodtech/portalserver/reference/techart/jsr168/

[13] WSRP Web Services for Remote Portlets http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrphttp://www.oasis-open.org

[14] Globus: http://www.globus.org

[15] GridLab: http://www.gridlab.org

[16] Liferay: http://www.liferay.com

[17] A. Akram,D. Chohan,X.Wang,X.Yang and R. Allan " A Service Oriented Architecture for Portals using Portlets", *All Hands On Meeting*, 2005

[18] The eXo platform: http://www.exoplatform.com

[19] B. Mestrallet, T. Nguyen et al "eXo Platform v2, Portal, JCR, ECM, Groupware and Business Intelligence". Available : http://www.theserverside.com/articles/article.tss?l=eXoPlatform

[20] StringBeans: http://www.nabh.com/projects/sbportal

[21] X. Yang, D. Chohan, X. Wang and R. Allan "A Web portal for National Grid Service" Presented at *GridShpere and Portlets workshop*, 03 March, 2006, eScience Institute, Edinburgh

[22] uPortal project: http://www.uportal.org/

[23] OGCE: http://www.ogce.org

[24] Pluto: http://portals.apache.org/pluto/

[25] Jetspeed-1: http://portals.apache.org/jetspeed-1/

[26] Jetspeed-2: http://portals.apache.org/jetspeed-2/

[27] IBM WebSphere Portal: http://www-306.ibm.com/software/genservers/portal/

[28] IBM Academic Initiative: http://www-304.ibm.com/jct09002c/university/scholars/

[29] RealityGrid: http://www.realitygrid.org/

[30] MyGrid project: http://www.mygrid.org.uk/

[31] GeneGrid: http://www.qub.ac.uk/escience/dev/article.php?tag=genegrid_summary

[32] P-GRADE grid portal http://www.lpds.sztaki.hu/pgportal/

[33] SAKAI VRE demonstrator http://tyne.dl.ac.uk/Sakai/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Integrating R into Discovery Net System

Qiang Lu, **Xinzhong Li**

{qianglu, xinzhong}@doc.ic.ac.uk
Dept. of Computing, Imperial College, 180 Queens Gate, London, SW7 2RH, UK

Moustafa Ghanem, Yike Guo

{mmg, yg}@inforsense.com
459a Fulham Road Chelsea,London SW10 9UZ, UK

Haiyan Pan

{hypan}@scbit.org
Shanghai Center for Bioinformation Technology, Shanghai, 210235, China.

Abstract

Discovery Net system, which is a workflow-based distributing computing environment, permits various tools to be integrated and thus provides a high-performance platform for data mining and knowledge discovering. As we know, the bioinformatics research field extends rapidly. Hundreds of various algorithms and software are developed every year. The unique capability of integration makes Discovery Net System an ideal uniform platform for bioinformatics research, where comprehensive and systematic analysis is always needed. As an open-source statistical tool, R is becoming very popular in bioinformatics research, especially in the field of microarray data analysis. Therefore, integrating R into Discovery Net system is of great significance. In this paper, we successfully developed a framework, with which R functions can be easily integrated into Discovery Net System without any further programming. Here, the methodology is illustrated, and an application instance for the domain of Microarray Analysis is demonstrated as well.

1. Introduction

Bioinformatics is a rapid growing research field, where thousands of papers are produced every year denoting new problems and new strategies in various aspects. Obviously, it is very hard for any software to handle all these problems, especially using up-to-date methodologies. Therefore, it is the normal case that some research centers and big pharmaceutical companies have several academic or commercial software systems even with overlap functions to help their daily data management and analysis. How to communicate between different systems actually becomes a challenge. The Discovery Net, one of EPSRC's six pilot projects (http://www.lesc.ic.ac.uk/) provides such an integrative analysis platform, Discovery Net System, which is a workflow-based distributing computing environment, permitting various tools to be integrated. The Discovery Net middleware is written in Java language with the technique of J2EE. In the system, every distributed (local or remote) algorithm/function is incorporated as a pluggable component (a node). A webstart enabled client provides a portal for user to access and manage the computation services. In the portal, not only functionalities for submitting task, retrieving result and managing workflow are provided, but also those for interactive visualization are included. Rowe A. et al. (2003), Jameel et al.(2004), and Curcin V. et al.(2004) have illustrated the Discovery Net System in detail and have demonstrated several application instances in genome research.

Recently, an open source program, R (http://www.r-project.org/) is becoming very popular in bioinformatics research. It provides a wide variety of statistical algorithms (linear nonlinear modeling, classical statistical tests, time-

series analysis, classification, clustering, micro-array analysis, and so on) and graphical facilities. Moreover, since R itself is free and high extensible, lots of life science projects in academics have been contributing to R, extending it to a lot of related areas quickly. So far, in R, not only its initial subject of statistics, but also some bioinformatics problems have been included and provided with exceptional efficient solutions.

Considering R's powerful functionality, some commercial data analysis packages such as GeneSpring, Spotfire and Rossetta Resolver, and some open source programs such as Gaggle, EBI Expression profiler, RACE (Psarros M. et al. 2005) have already integrated R into their latest versions.

Here we introduce the integration of R into Discovery Net System, as well as its usage on the field of microarray expression data analysis. The most significant point of this integration of R is that, a generic framework is provided, with which no further programming is needed when user integrates R functions or subroutines by themselves. The integration can be done in a couple of minutes.

2. Method

R is the language and environment having two significant functionalities: 1) statistical computing and 2) interactive graphics. Having R in Discovery Net System, naturally, not only the powerful statistical algorithms but also the interactive graphical facilities is expected.

2.1 Integration of Algorithms

In general, algorithms in R are described in S language. R Software itself provides an engine to execute these scripts, and a GUI to interact with users. To integrate its algorithms, viz. invoke R functions from Discovery Net System, R API is used, which is written in C language and provided as a share library. Since the Discovery Net System framework is built with Java language, Java Native Interface (JNI) is introduced to fill the gap in invocation stack, as showed in Figure 1.

Standing beside the invocation stack, the more significant point is the generic framework, General-R, which facilitates the integration. The main idea of this generic framework is a XML file, viz. i3xml (XML implemented interface 3 of WfMC Workflow Reference Model), which has the format of Web Services Description Language (WSDL), describing the R functions to fit node specification of Discovery Net System.

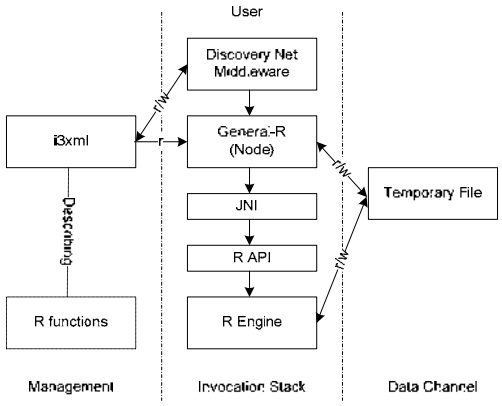


Figure.1 Architecture of Bridging Discovery Net and R, where r/w means read and write respectively.

Using WSDL, an R function can be described, the input and output parameters as messages, while the functions (name) as operations.

With the support of XML Schema Definition (XSD), the types of input and output messages are flexible. Not only some basic types such as Double, Integer, String, and Boolean, but also some complex types such as array and matrix can be used. Moreover, to transfer raw dataset, a special data type, REXP (R EXPression), is introduced.

To operations, three parts, init, action and final are designed to describe R function. They are supposed to be invoked by R engine in succession. For example as that in table 1, the scripts of init.R and final.R under the folder of preprocessAffy/expresso are invoked before the action of expresso().

```
<wsdl:operation name="Expresso">
   <kde:init  script="preprocessAffy/expresso/init"/>
   <kde:final script="preprocessAffy/expresso/final"/>
   <kde:operation action="eset_tmp&lt;-
expresso(abatch,bgcorrect.method=bgcorrect,normalize.metho
d=normalize,pmcorrect.method=pmcorrect,summary.method=
summary)"/>
</wsdl:operation>
```

Table 1 Example of Operation, where Bioconductor function Expresso is integrated.

After the i3xml gives the description of R functions, what the framework of General-R does is to read and interpret the description in the i3xml file, and thus invoke R engine to execute corresponding R scripts.

In Discovery Net System, the components such as those for algorithms are specified with input data/metadata, output data/metadata and parameters. A run task in Discovery Net System is organized as a workflow. Theoretically, the input

is from previous component, while the output is to the next component. The parameters are typed-in before execution. To describe an R function as a Discovery Net System component, inputs, outputs and parameters need to be defined. Generally, the inputs, outputs and parameters are related to the input and output messages defined in the i3xml file.

Obviously, to edit the XML file by text editor is an exhausting work. Considering this, an editor for i3xml file, XML Wrapper, is developed.

2.2 Integration of Interactive Graphics

With the above integration of algorithm, any graphical results created by R functions can be obtained by defining the result with graphical type such as JPGPicture, PostscriptPicture or PNGPicture. However, this method means the properties of graphics such as axes, color, and legend etc. should be defined in advance. Obviously it is not convenient enough, especially for exploring the dataset interactively.

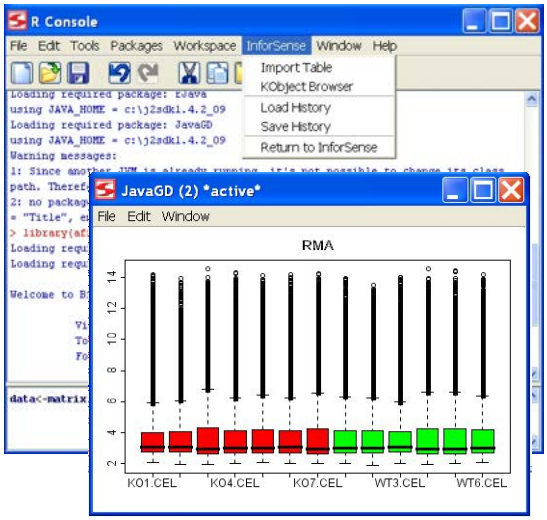


Figure 2 Integration of Inteactive Graphics

Considering the above disadvantage, R is integrated into Discovery Net System client as well. To facilitate the integration, the 3rd party software, JGR, is adopted. JGR (Java Gui for R) includes interactive interface to edit and input Java script. It uses a Java graphics device, JavaGD, in which all painting functions of R are delegated to the Java class. After add some functions for communicating between Discovery Net System and R, such as loading data from and saving figures to Discovery Net System, the function of interactive graphics is implemented as Expert-R node, as Figure 2 shows.

3 Application

Wellcome Trust Functional Genomics Development Initiative funds a programme: Biological Atlas of Insulin Resistance (BAIR, http://www.bair.org.uk) trying to address the mechanism of insulin resistance. Discovery Net System is chosen to be the main platform for its daily data analysis.

Here, we give an example of how to deal with IRS2 knockout mice expression data by the R integration, which is a part of BAIR research. Figure 3 gives the analysis workflow.

There are 7 IRS2 knockout and 6 wildtype Affymetrix MOE430v2 chips in this application. We chose standard RMA normalization approach. MAS5 present/absent call is calculated as well. Data is 2-based logarithm transformed. Multiple test was applied for FDR test after Welch's t-test. The predefined Bonferroni and Benjamini & Hochberg test and Storey's FDR in R multi-test package were performed.

After applying Benjamini & Hochberg FDR test, we found that no probeset can pass the test of FDR < 0.05, while only 9 probesets pass that of FDR<0.25. Among these 9 probesets, 3 probesets are absent cross all 13 chips, one probeset does not change enough. (FoldChange =1.03). The other 5 genes, Gpd2, Atp1b1, Pak1ip1 and Cdkn1b about 30% down regulated, and Dnpep about 60% up regulated, remain significantly expressed between IRS2 knockout and wildtype mice. Among these five genes, *Cdkn1b* described as protein p27(Kip1), regulates cell cycle progression in mammals by inhibiting the activity of cyclin-dependent kinases (CDKs). It is confirmed by other researchers (T. Uchida, 2005) that deletion of *Cdkn1b* ameliorates hyperglycemia by maintaining compensatory hyperinsulinemia in diabetic mice, thus, p27(Kip1) contributes to beta-cell failure during the development of type 2 diabetes in Irs2 knockout mice and represents a potential new target for the treatment of this condition.

Because nearly no single gene passed multiple comparison, it's necessary to identify groups of genes with similar regulation between IRS2 knockout and wildtype. First we got 586 probesets (represent 524 unique genes) by welch's t-test < 0.01, then by using Onto-Express (http://vortex.cs.wayne.edu/ontoexpress/) with multiple correction testing, which indicated that RNA binding (P=0.0), endocytosis (P=5.0E-5), protein modification (P=3.3E-4), nucleus (P=5.4E-4), perimuclear region (P=6.8E-4), vesicle-

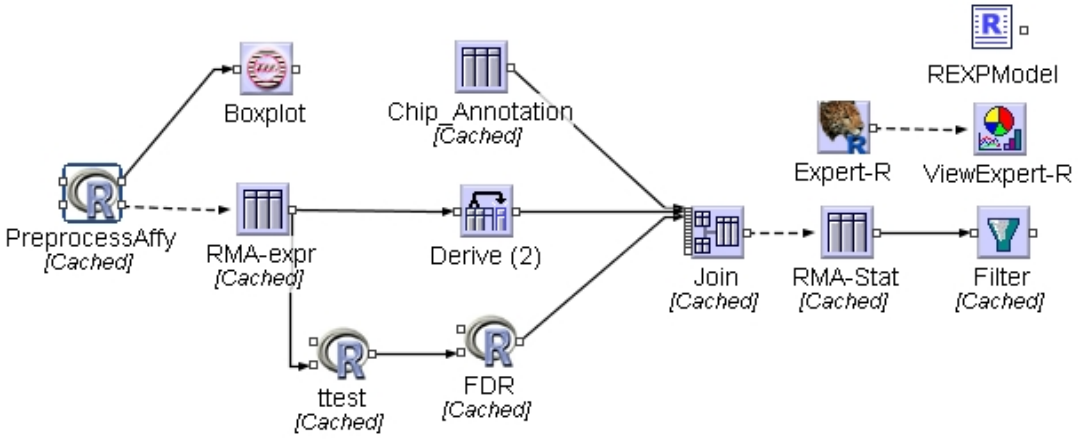


Figure 3 Workflows with R integration for BAIR, where ProprocessAffy, ttest, FDR are nodes with General-R framework, while Expert-R is the interactive R environment integrated. The ViewExpert-R is the figures created. Other nodes, such as Boxplot, Filter, Derive, Join are some other Knowledge Discovery Network nodes.

mediated transport (P=0.002) and lipid biosynthesis (P=0.003) ontology groups were represented to a significantly greater extent in wildtype vs. IRS2 knockout than expected if ontology groups were randomly distributed within the list of these 524 top genes.

4. Summary

As we have seen, with the general-R framework we have developed, the user can easily integrate the R functions or subroutines into Discovery Net System. So far, many R functions in package Bioconductor have been integrated, which empowers the Discovery Net System for micro-array analysis.

However, some improvements are proposed. Firstly, instead of using XML editor to integrate, the method of annotation is proposed. With adding annotations into R code as comments, the input, output messages and operations are defined. The general-R framework will parse R script to get the integration information including message name, type, and operation etc. and thus integrate them into Discovery Net System automatically.

Secondly, with the introduction of annotation, the client-side expert-R will act as not only an interactive graphics tool, but also a console for integration and debug environment. Finally, more popular R packages/functions will be integrated.

Acknowledgements

The authors would like to thank all other members working for the project of Discovery Net. Thanks to Dr. Alex Michie in Inforsense Ltd for good suggestions. Thanks to BAIR project funded by Wellcome Trust, and Discovery Net project funded under the UK e-Science Programme.

Reference

1. Curcin V., Ghanem M., Guo Y., et al., (2004) SARS Analysis on the Grid. http://www.allhands.org.uk/submissions/papers/80.pdf
2. JGR, http://stats.math.uni-augsburg.de/JGR/
3. Psarros M., Heberl S., Sick M., et.al., (2005) RACE: Remote Analysis Computation for gene Expression data, Nucleic Acids Res. 33 (Web Server issue):W638-43.
4. Rowe A., Kalaitzopoulos D., Osmond M., et al., (2003), The discovery net system for high throughput bioinformation, Bioinformatics, 19 (supp), i225-i231.
5. Syed J., Ghanem M., and Y. Guo (2004), Discovery Processes in Discovery Net, http://www.allhands.org.uk/submissions/papers/110.pdf.
6. Uchida T., Nakamura T., Hashimoto N., et al., (2005) Deletion of *Cdkn1b* ameliorates hyperglycemia by maintaining compensatory hyperinsulinemia in diabetic mice. Nat Med., 11(2):175-82.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Cardiff University's Condor Pool: Background, Case Studies, and fEC

James Osborne[1], Alex Hardisty[2]

1. Information Services, Cardiff University, 50 Park Place, Cardiff, CF10 3AT, Wales UK
2. Welsh e-Science Centre, Cardiff University, 5 The Parade, Cardiff, CF24 3AA, Wales UK

Abstract

Cardiff University's Condor pool aims to cater for the high throughput computing needs of a wide range of users based in various schools across campus. The paper begins by discussing the background of Cardiff University's Condor pool. The paper then presents a selection of case studies and outlines our strategy for Condorising applications using submit script generators. Finally the paper presents the findings of a fEC exercise and outlines our policy for attributing directly allocated costs to a particular research project.

1. Background

Cardiff University's Condor pool is the third largest Condor pool in the UK with an average of around 800 execute nodes providing 500 gigaflops of computing power on-demand to our researchers, giving them a significant advantage over their competitors in other universities whilst at the same time saving between a quarter and a third of a million pounds on dedicated high throughput computing resources [1].

Cardiff University's Condor pool began as a pilot service back in April 2004 as an initiative sponsored by Information Services and the Welsh e-Science Centre after Dr Hugh Beedie of Information Services realised the machines they were using to provide an open access workstation service to support learning and teaching were being under utilised.

In April 2004 Condor 6.6.2 was deployed on a number of open access workstations running Windows NT using ZENworks application management tools. Condor 6.6.2 was configured and tested on a dedicated test cluster by Jon Giddy and Chris Tottle prior to deployment.

One of the applications Condorised during the pilot phase involved a colleague in the School of Biosciences, Dr Kevin Ashelford. His application used the university's Condor pool to perform a distributed search to identify corrupt records in a DNA database. Kevin would have had to run his application constantly meaning the elapsed wall-clock time would have been of the order of 28 months. Using Condor Kevin ran his application in parallel reducing the time taken to 18 days.

In April 2006 Condor 6.6.11 was deployed on a number of open access and schools donated workstations running Windows XP using ZENworks application management tools. Condor 6.6.11 was also configured and tested on the dedicated test cluster, this time by Dr James Osborne prior to deployment. James was responsible for tightening up the security policy and defining an additional job control policy for long running jobs.

All applications that run on the open access workstations are tested on the dedicated test cluster prior to allowing them to run on the production system and long before we allow users to run their own applications allowing us to reduce the risk of applications causing problems on the production system.

2. Case Studies

The three most significant research projects in terms of their potential or actual consumption of Condor hours are presented in this section.

2.1 The School of Computer Science in Collaboration with Velindre Cancer Centre

BEAMnrc and DOSXYZnrc are applications that use Monte Carlo simulation to perform radiotherapy dose calculation [2].

Before Condorising these applications our colleagues in the School of Computer Science, Prof David Walker, and at the Velindre Cancer Centre, Dr Geraint Lewis and Mary Chin, were already investigating grid computing methods to reduce the time taken to simulate a single radiotherapy dose calculation.

Mary would have had to run BEAMnrc and DOSXYZnrc long enough to simulate 45 to 60 million X-ray events per radiotherapy dose calculation, each dose calculation taking between 2,430 and 3,240 hours to complete. Mary would have had to run BEAMnrc and DOSXYZnrc constantly meaning the elapsed

wall-clock time to simulate a single dose calculation would have been of the order of 3.5 to 4.5 months. In the life of a cancer patient months may be too long to wait hence the investigation of grid computing methods.

Using Condor Mary can run BEAMnrc and DOSXYZnrc in parallel reducing the time taken to simulate a single radiotherapy dose calculation to 36 hours.

During the first quarter of 2006 Mary used a total of 2,771 Condor hours which equates to 8% of the total Condor hours consumed. During the second quarter she used a total of 5,474 Condor hours which equates to 6% of the total Condor hours consumed. If we were charging Mary in line with our external fEC policy we would have charged her £55 in the first quarter and £109 in the second.

The Condorisation of BEAMnrc and DOSXYZnrc was performed by Mary Chin who is based at Velindre Cancer Centre.

2.2 The School of Biosciences

Structure is an application that uses Bayesian classification to assign individual genetic sequences to appropriate populations [3,4].

Before Condorising Structure our colleague in the School of Biosciences, Tim Bray, would typically run Structure on the same individual genetic sequence a total of twenty times, each run taking 12 hours to complete. Tim used to run Structure overnight meaning the elapsed wall-clock time to assign an individual genetic sequence to an appropriate population used to be of the order of 4 weeks.

Using Condor Tim can run Structure in parallel reducing the time taken to assign an individual genetic sequence to an appropriate population to 12 hours. In addition Tim can run Structure on multiple genetic sequences at the same time. Tim said "Condor has allowed submission of jobs in large numbers that will run in parallel as long as there are free machines available." Tim also said "In this way an analysis that may take a day for a single sequence has the potential to be finished with twenty repeats in the same period."

During the first quarter of 2006 Tim used a total of 5,915 Condor hours which equates to 5% of the total Condor hours consumed. During the second quarter he used a total of 15,221 Condor hours which equates to 17% of the total Condor hours consumed. If we were charging Tim in line with our internal fEC policy we would have charged him £89 in the first quarter and £228 in the second.

The Condorisation of Structure was performed by Steffan Adams who is based in the School of Biosciences. Steffan is responsible for a small Linux based Condor pool used by the school. Steffan ported his solution to the university's Windows based Condor pool because Tim was saturating the school's Condor pool.

2.3 The School of Optometry and Vision Sciences

Dammin and Gasbor are applications that use ab initio methods to build models of proteins using simulated annealing [5,6].

Before Condorising these applications our colleagues in the School of Optometry and Vision Sciences, Prof Tim Wess and Donna Lammie, would typically run Dammin or Gasbor on the ab initio data of a particular protein a total of twenty times, each run taking either 18 or 90 minutes to complete using Dammin or Gasbor respectively. Donna used to run Dammin or Gasbor during the day meaning the elapsed wall-clock time to build a model of a protein used to be of the order of either 1 or 4 days using Dammin or Gasbor respectively.

Using Condor Donna can run Dammin or Gasbor in parallel reducing the time taken to build a model of a protein using Gasbor to 2 hours. In addition Donna can run Dammin or Gasbor on multiple ab initio datasets at the same time. Donna said "Condor has proved invaluable to our research since the work is completed rapidly and efficiently."

During the first quarter of 2006 Donna used a total of 1,876 Condor hours which equates to 2% of the total Condor hours consumed. During the second quarter she used a total of 25,566 Condor hours which equates to 28% of the total Condor hours consumed. If we were charging Donna in line with our internal fEC policy we would have charged her £28 in the first quarter and £383 in the second.

The Condorisation of Dammin and Gasbor were performed by James Osborne who is based in Information Services. James is responsible for the university's Condor pool. James Condorised Dammin and Gasbor by writing a program to generate Condor submit scripts. Developing the submit script generator was done in such a way so that the generator could be quickly adapted to support other applications in the future. To date the generator has been adapted to support five additional applications. We briefly discuss the submit script generator in the next subsection.

2.4 Condorising Applications Using Submit Script Generators

The submit script generator allows us to rapidly Condorise applications concerned with data processing. We will use the Dammin and Gasbor applications to aid our discussions.

Dammin and Gasbor can be used to build models of proteins using simulated annealing. The input files for Dammin and Gasbor are produced by an application called Gnom which is used to filter the data captured by the experimental apparatus which bombards a sample of the protein under investigation with X-rays.

Dammin and Gasbor were originally designed as interactive applications asking the user a number of questions before processing the output from Gnom and generating a model of the protein that can be visualized. Dammin and Gasbor can also operate in batch processing mode by providing an answer file containing the answers the user would otherwise have had to supply during interactive mode.

When running Dammin, Donna would typically accept all the default answers to the questions Dammin asked except the name of the Gnom file, the name of the log file (used to log any errors), and the name of the project identifier (used to name the output file).

Using the submit script generator Donna does not have to answer a single question, the generator simply looks in the input directory for the Gnom file and generates twenty answer files containing the name of the Gnom file, the name of the log file in the format file0.log to file19.log, and the name of the project identifier in the format file0 to file19. The submit script generator then generates twenty batch files calling Dammin with one of the twenty answer files.

The submit script generator then builds the Condor submit script itself which in turn transfers copies of the Dammin binary, the Gnom file, the answer file, and the batch file to each execute node and tells Condor where to transfer the output files to on the submit node. The script generator is also capable of processing multiple Gnom files in the input directory.

When running Gasbor, Donna would typically accept all the default answers to the questions Gasbor asked except the name of the Gnom file, the name of the log file, the project identifier, and the number of residues in the asymmetric part.

Using the submit script generator Donna only has to answer one question, the number of residues in the asymmetric part. The Gasbor submit script generator works in the same way as the Dammin submit script generator.

3. Full Economic Costing

A full economic costing of Cardiff University's Condor pool was conducted in line with various higher education funding council's requirements for full economic costing of research projects.

3.1 Terminology

The full economic costing (fEC) model divides costs into two types, indirect costs and direct costs. The model further divides direct costs into two subtypes, directly incurred costs and directly allocated costs [7].

Indirect costs are those costs incurred that are not directly related to any one project but costs that are necessary to support a given project. Directly incurred costs are those costs incurred for equipment or services related to a single project. Directly allocated costs are those costs incurred for equipment or services shared by a number of projects.

3.2 Indirect Costs

The indirect cost of the university's Condor pool is the cost of equipment, power, and staff required to provide execute nodes, submit nodes, and networking which are already provided by Information Services as part of the university's overall costs in providing an open access workstation service for students to support learning and teaching.

The cost of the open access workstation service includes: initial purchase of machines with three-year warranties updated on a four-year rolling cycle, the cost of power consumed, the cost of support staff required to maintain and update the machines, and an element of cost for networking and central data storage.

The indirect cost of the university's Condor pool is recovered via the indirect charge in £ per FTE per year added to the full economic costing of every research project.

3.3 Direct Costs

The direct cost of the university's Condor pool is the cost of equipment, power, and staff required to provide the central manager and Condor support services beyond those provided by Information Services as part of the open access workstation service.

The direct equipment cost is the cost of the central manager which includes: initial purchase of the machine with a three-year warranty

updated on a four-year rolling cycle, as well as an annual racking fee that includes: rack space, uninterruptible power supply, air conditioning, and network connection. The cost of the central manager is £1,560 pa.

The direct power cost is significantly less than the cost of running the pool at maximum capacity which, based on current market prices, would be £49,640 pa.

The direct staff cost is currently £35,000 pa which includes one member of full-time staff and an element of cost for administration.

Currently the direct cost of the university's Condor pool is met by Information Services and the Welsh e-Science Centre. In the future the direct cost of the university's Condor pool will become a directly allocated cost shared between the research projects using the Condor service.

We briefly discuss how to attribute directly allocated costs to a particular research project in the next subsection.

3.4 Attributing Directly Allocated Costs to Particular Research Projects

Directly allocated costs can be attributed to a particular research project using the accounting information collected by the central manager.

A value for directly allocated costs can be calculated by dividing the maximum directly allocated costs of the Condor pool by the maximum number of Condor hours available which gives us a cost of 1.5 pence per Condor hour.

Calculating the directly allocated equipment and staff costs is trivial, however calculating the power cost is a little more involved. To do this we measured the power consumed by a number of different open access workstations of various specifications.

We measured the power consumption of each sample machine in three different states for a total of fifteen minutes. A sample machine in the first state, IDLE, was simply turned on with nobody logged in. A sample machine in the second state, MAX CPU, was running a Condor job that called the CPUSoak program provided with the Condor toolkit. A sample machine in the third state, MAX DISK, was busy copying and deleting an ISO file over and over.

We then calculated the combined additional power consumption of each sample machine whilst running at MAX CPU and MAX DISK.

We then used census information to calculate the cost of running the pool for an hour and divided that by the number of machines in the pool giving us an average cost of 0.5 pence per Condor hour.

We recommend that future research projects that wish to use the university's Condor pool try to estimate, with our assistance, the number of Condor hours needed to satisfy their computational requirements in order that directly allocated costs can be included and subsequently recovered via the fEC of their own research projects.

Our internal charging policy is to charge 1.5 pence per Condor hour whereas our external charging policy is to charge 2.0 pence per Condor hour.

Acknowledgements

We are grateful to those users who provided case studies: Steffan Adams, Kevin Ashelford, Tim Bray, Mary Chin, Donna Lammie, Geraint Lewis, David Walker, and Tim Wess. We are also grateful to Information Services and the Welsh e-Science Centre for funding this research.

References

[1] M. Litzkow, M. Livny, and M. Mutka. Condor – A Hunter of Idle Workstations. In *Proceedings of 8th IEEE International Conference on Distributed Computing Systems 1988 (ICDCS8)*, pages 104-111, San Jose, California, USA, June 1988. IEEE.

[2] M. Chin, G. Lewis, and J. Giddy. Implementation of BEAMnrc Monte Carlo Simulations on the Grid. In *Proceedings of 14th International Conference on the Use of Computers in Radiation Therapy 2004 (ICCR2004)*, Seoul, Korea, May 2004.

[3] J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of Population Structure Using Multilocus Genotype Data. *Genetics*, 155(2):945-959, 2000.

[4] G. Evanno, S. Regnaut, and J. Goudet. Detecting the Number of Clusters of Individuals Using the Software Structure: a Simulation Study. *Molecular Ecology*, 14:2611-2620, 2005.

[5] D. I. Svergun. Restoring Low Resolution Structure of Biological Macromolecules from Solution Scattering Using Simulated Annealing. *Biophysical Journal*, 76:2879-2886, 1999.

[6] D. I. Svergun, M. V. Petoukhov, and M. H. J. Koch. Determination of Domain Structure of Proteins from X-Ray Solution Scattering. *Biophysical Journal*, 80:2946-2953, 2001.

[7] The Joint Costing and Pricing Steering Group. Internet. http://www.jcpsg.ac.uk/, July 2006.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

A Peer-to-Peer Architecture for *e*-Science

Stratis D. Viglas
School of Informatics, University of Edinburgh, UK
sviglas@inf.ed.ac.uk

Abstract

One of the key issues in supporting *e*-Science is managing data in distributed, flexible, scalable and autonomous ways. This paper proposes an architecture based on Peer-to-Peer systems that can be used to facilitate large-scale distributed data management. It identifies the important problems that need to be addressed and presents possible solutions, along with their expected advantages. These ideas, we believe, are helpful in extending the discussion of alternative approaches of supporting the multiple facets of *e*-Science.

1 Introduction

Research organisations produce data at ever-increasing rates. Central management is impossible for a variety of reasons, including, but not limited to, the sheer volume of data, their rate of change, and their geographical distribution. This means that flexible and strictly distributed architectures need to be in place. The purpose of this paper is to present such an architecture with an increased focus on decentralised, scalable and reliable data management. That is not to say that high performance issues are to be discarded; rather, by focusing on reliable data management we can "free" the applications built on top of the data management layer to concentrate on application-specific performance issues without having to address management aspects as well.

Large-scale decentralisation. Accumulating information in central structures means that central points of failure are created, and fault-tolerance is decreased. The situation can be alleviated by employing overlay networks (*e.g.*, [17, 18, 19]) but, even in those cases, targeted attacks can still take place, while secondary maintenance protocols need to be continuously executed to keep the overlay structure updated. We would like the system not to have any rigidly imposed structure, but be fully adaptable.

Increased autonomy. In a decentralised system, nodes join and leave at will. There are no established "contracts" as to how long a node should be in the system, or of replication of the data available at a single node. The node itself manages its behaviour, along with access to the data it serves. This poses questions as to what connections the node establishes and what protocols are to be executed upon node arrivals and departures.

Security. Naturally, there is need for security, especially if there is no central management authority. We focus on user-level data access, so as not to (*i*) compromise data integrity, or (*ii*) allow access to users who the nodes of the system do not want to grant access to. Both these aspects are in direct accordance to the autonomy notion previously described.

Efficient data retrieval and manipulation. Performance in a decentralised system is of paramount importance; in our setting, performance means response time. Current research has addressed the problem by either (*i*) building high-bandwidth connections and relying on the speed of those connections to account for rapid data exchange, or (*ii*) building overlay networks and measuring performance in terms of the number of routing hops necessary to route data retrieval requests. Both metrics provide localised solutions to a truly global problem in a decentralised data management system. For instance, rapid data retrieval does not account for what data manipulation takes place over said data; or, the number of routing hops to locate data does not take into account data volume, or network latency. Being agnostic to the rest of the computing environment is not helpful in a decentralised data management scenario, especially one as intricate and collaborative as *e*-Science.

We aim to address all these issues by developing customisable middleware between the nodes comprising the system. Participating nodes will need only implement a specific interface, therefore being independent of any ties to operating systems or programming environments. In light of the needs for autonomy and decentralisation, we propose a Peer-to-Peer (P2P) architecture. In the following sections, we shall present such an architecture, focus on a subset of the problems at hand, and present possible solutions.

2 System Overview

The general overview of the system is shown in Figure 1. The main assumptions are: (*i*) Each peer exports its data in XML. This imposes no restriction on the peer's native data format; the only requirement is that an XML view of it is exported. (*ii*) Each new peer is introduced to the system by following a "hand-shake" protocol with an existing peer. (*iii*) Each peer maintains connections to other peers, forming its *routing table* that contains both semantic and structural links. (*iv*) While present in the system, the peer's routing table evolves to account for peer arrivals or departures. (*v*) To process queries, a peer first identifies the peers relevant to the query. The query is then rewritten in ways that can be processed by the relevant peers. (*vi*) A departing peer executes a specialised exit protocol.

2.1 Protocols

The functionality of the system can be summarised in three main protocols. The implementation of these protocols dictates additional building blocks of the architecture.

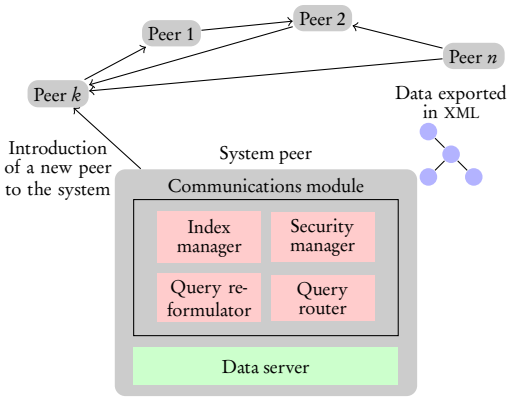


Figure 1: System overview and general architecture

Entrance protocol. This is the "bootstrapping" sequence executed by a peer joining the system. The requirement is that a joining peer knows of another peer that is already part of the system. Consider, *e.g.*, peer P_n joining the system and being aware of peer P_o. There are two steps: (*i*) exporting of P_n's data and indexing of its data at P_o, and (*ii*) exchange of information between the two peers concerning further nodes. The first step introduces P_n to the system and makes it accessible by peers known to P_o, *i.e.*, P_o is now capable of delegating queries to P_n. The second step accounts for the reverse direction, *i.e.*, for P_n to be able to route queries to P_o; P_o becomes part of P_n's routing table and vice-versa. These steps can be recursively applied: P_o can forward P_n's credentials to peers it knows about and it can forward the credentials of those peers to P_n.

Querying protocol. Each peer should be in a position to retrieve data served by any other peer in the system. The querying protocol undertakes the following: (*i*) identifying the peers relevant to a query; (*ii*) translating the query to a destination peer's exported schema; (*iii*) forwarding the query to the destination peer and reporting the results back to the user; and (*iv*) while answering the query, update the originating peer's routing table with newly discovered information. These steps are not executed only at a single peer. Since multiple peers may be useful in answering a single query, the steps are recursively applied by all participating nodes. This forms the basis of *data discovery* and *query routing*, two salient features in decentralised indexing and query evaluation.

Exit protocol. On exiting, a peer disassociates itself from peers it knows about. Assuming peer P_n leaves the network, during the exit protocol, it: (*i*) propagates information it has gathered during its stay in the system to peers it knows about, so as for any associations it has identified to "live on," and (*ii*) makes its data inaccessible to the peers it is connected to, so as to bring the system to a consistent state. As before, the first step can be recursively applied.

2.2 Modules

To realise the previous protocols, each peer has two modules: (*i*) a data server, responsible for exporting data to XML and locally evaluating queries, and (*ii*) a communication module, providing access to remote peers of the system (see also Figure 1). The responsibilities of the communication module are delegated to four entities, described next.

Index manager. The index manager is responsible for maintaining data connections between related peers. These connections are either (*i*) *semantic*, representing related concepts (*e.g.*, peers that maintain information about related scientific data may be aware of each other), or (*ii*) *structural*, used when reformulating and routing queries to relevant peers. Semantic connections are stored as mappings, while structural ones are stored as communication links.

Security manager. Autonomy is one of the important aspects of the architecture, *i.e.*, a peer should be responsible for both serving data, as well as controlling access to it. This means that security policies need to be in place. In the spirit of decentralisation, this information needs to be completely distributed throughout the system.

Query reformulator. The query reformulator is responsible for rewriting queries before forwarding them to remote peers. It "translates" the query at hand to use terms that are known to the destination peers. For instance, if a query is to be routed to institutions using different words for the same term, the query reformulator consults local mappings and forwards the query rewritten in ways that can be processed by the remote peers.

Query router. After relevant peers have been identified and the queries have been reformulated, the question is how should these queries be evaluated. Query performance characterises system performance, so it is crucial to be addressed in an efficient manner. In a dynamic and decentralised system, local optimisation decisions may prove quite limiting. We propose query evaluation through routing; the entity routed can either be a part of the query, or a partial result of the query, or even the entire query if this is deemed the best evaluation strategy.

3 Research Issues

We now turn to the core research agenda of our proposal, decomposed into four major categories: indexing, security, query reformulation and query evaluation.

3.1 Decentralised Indexing

Each peer autonomously manages and extends its own routing table to be used during query evaluation. The general form of this index structure is shown in Figure 2a. A local routing table is conceptually a three-column relational table. The first column is a destination peer, the second a local term and the third column a remote term (*i.e.*, a part of the schema exported by the destination peer). As shown in Figure 2a, a peer's routing table contains both semantic and structural information. Any entry of the routing table contains structural information; semantic information comes into play by maintaining *term mappings* wherever that is applicable. For instance, the first row in Figure 2a's routing table means that local term X maps to remote term A. Note that the same peer (*e.g.*, Peer 1) may appear multiple times in the routing table. In addition, the same local term may be mapped to multiple remote

terms at different peers (*e.g.,* Peers 1 and i in Figure 2a). Finally, the same remote term may be served by multiple peers (*e.g.,* for Peers 1 and n in Figure 2a). This give us substantial flexibility in identifying relevant peers or even choosing between alternative peers serving related information.

The questions that arise have to do with forming and maintaining routing tables: (*i*) How much information is exchanged whenever a new peer joins the network? Alternatives include the new peer "downloading" the existing peer's entire routing table, or a part of it. (*ii*) How is the routing table updated as peers join and leave the network? One option is to "piggy-back" the routing table updates when accessing remote peers for the purposes of querying. Another option is to have a maintenance protocol being executed periodically. (*iii*) How can the routing table be further utilised during query evaluation? In particular, can the maintenance protocol provide performance guarantees about the connections stored in the routing table (*e.g.,* latency, probability of the connection being up to date *etc.*). Decentralised indexing allows for a great deal of research to be undertaken in the area.

3.2 Security

Data management means that, in addition to serving, peers manage access to the served data. The question that emerges is one of security: how can peers control which peers access their data? One solution is for each peer to request each other peer to register with it. However, this solution will not scale: it goes against the idea of complete decentralisation (as it means that each peer is aware of all peers in the system), while it also inhibits the maintenance protocols described earlier (as even the slightest local changes need to be globally transmitted). Additionally, we would like all forbidden requests to *fail fast*, *i.e.,* to fail as soon as possible – ideally at the originating peer. This means that each peer not only keeps track of who has access to its data, it also maintains information about what data it has access to.

The solution we propose is based on XML security views [8]. Each peer exports different views of its data to different peers, depending on the peer it is communicating with. An example is shown in Figure 2b where Peer k, exports different views to Peers i and j. The system adheres to the fail fast principle: since neither Peer i nor Peer j are aware of the data they cannot access, they cannot request it.

3.3 Query Reformulation

Query reformulation can be thought of as the query compilation step in a decentralised system. During query reformulation the system executes a *resource discovery* protocol, which aids in: (*i*) identifying peers that may contain relevant terms; (*ii*) translating the query to terms that are understandable by other peers; and (*iii*) ensuring that each requesting peer has access to a particular term by consulting security policies.

The three steps mentioned above are iteratively executed. For instance, in Figure 2a, if a query about term X is received, the local peer knows that in addition to accessing its local data, it needs to forward an appropriate query to Peer 1. The query is formulated by translating term X in the query to term A for Peer 1 to be able to handle it. In addition, the same procedure can be undertaken once the reformulated query reaches Peer 1 and for term A. The outcome of this process is a path $(P_1, Q_1)/(P_2, Q_2)/\ldots/(P_n, Q_n)$ with the semantics that at each peer P_i the corresponding reformulated query Q_i should be evaluated.

The path may indeed contain duplicate peers, *i.e.,* the same peer may have to be visited multiple times in processing a query. The most efficient way of accessing such peers is an issue of query optimisation and evaluation and is the purpose of the query router module.

3.4 Query Evaluation

An integral part of query evaluation is query optimisation. Already a hard problem in centralised environments, the situation is aggravated in decentralised ones as the likelihood of optimisation-time assumptions holding during evaluation-time is even lower. We propose decomposing queries into a query algebra and reducing query evaluation to a routing problem. We shall use two basic routing/evaluation strategies, shown in Figures 2c and 2d. The first is *parallel evaluation*: data requests are forwarded to peers, which then upload their data to the requesting peer that locally evaluates the query. The alternative is *serial evaluation*: a route is established and all peers are visited serially until the complete result is produced.

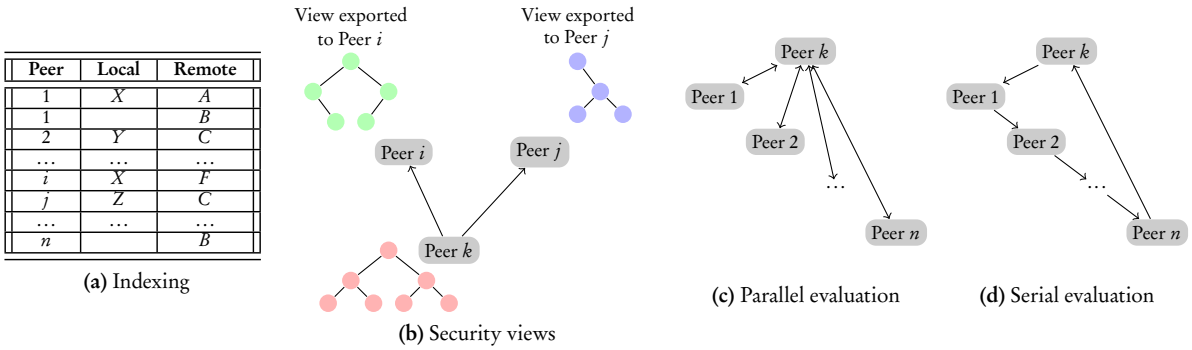
The two approaches can be better explained through an example. Consider a relational query of the form $R_1.a_1 = R_2.a_2 = \ldots = R_n.a_n$ posed at peer P_k where each R_i resides on a different peer P_i of the system. The parallel strategy would send requests for each R_i to be transmitted to P_k and for P_k to locally evaluate the query. The serial strategy instructs that the query be sent to P_1, the peer responsible for R_1, which then rewrites the query[1] and forwards it to P_2, the next peer in the sequence. The process is repeated until P_n is reached, which then sends the result to the originating peer P_k.

Note that the original query can be rewritten in many ways. In the previous example the original query can be decomposed into numerous blocks[2]; each block can be evaluated in parallel or serially. Each peer makes local routing, and, hence, optimisation decisions. Finally, note that a single peer may need to be visited multiple times. In such cases, care needs to be taken so that the number of times a single peer is visited is minimised.

To summarise, we envision query evaluation through query routing to be a continuous cycle of (*i*) mapping the query to a query algebra and forming query blocks, (*ii*) performing local optimisation to identify whether parallel or serial evaluation is more beneficial for a particular block, (*iii*) forwarding a query block to the peers storing data relevant to the block, and (*iv*) rewriting the query to

[1] A simple, but most likely inefficient, rewrite would be to substitute values in $R_1.a_1$ with constants.

[2] $\sum_{i=1}^{n} i!\binom{n}{i} = \sum_{i=1}^{n} \frac{n!}{(n-i)!}$ to be exact.



(a) Indexing

(b) Security views

(c) Parallel evaluation

(d) Serial evaluation

Figure 2: Various aspects of the proposed architecture

adjust for the partially computed result. The metrics to be employed in such an evaluation paradigm stem from three directions: (*i*) number of routing steps needed to evaluate the query; (*ii*) number of times a single peer is visited in evaluating the query; and (*iii*) raw size of data exchanged. Optimising for different dimensions, or combining all three metrics into a single one presents a very interesting multi-objective optimisation problem.

4 Related Work

There has been a host of work on P2P overlay networks and decentralised data structures in general (*e.g.,* [1, 3, 4, 6, 10, 16, 17, 18, 19]). These aim to solve the problem of efficiently identifying the peers of a system responsible for some particular data item by implementing a dictionary interface. In our case, the objective is to have highly unstructured networks that evolve as peers join and leave.

Another large area of work is concerned with distributed catalogs and covers peer data management systems (*e.g.,* [13, 20, 21]) and semantic overlay networks (*e.g.,* [5, 11]). In terms of query evaluation, there has been plenty of work on parallel databases (see *e.g.,* [7]) and distributed query processing (see *e.g.,* [14]). These approaches address environments of rigid structure and high predictability, while later studies have focused on the unpredictable behaviour of P2P systems (*e.g.,* [2, 9, 12]). All these focus on specific sub-problems without proposing a single, modular framework that is conducive to *e*-Science.

Finally, though *e*-Science oriented, existing approaches like the OGSA-DAI framework [15] address the problems at highly structured environments, without addressing intermittent peer behaviour or differing security policies. Rather, they focus on data integration and data delivery over Grid-like environments. It is certainly interesting to explore collaboration between the two approaches.

5 Conclusions

We have presented an architecture for building scalable data management systems over Web Services. We have focused on presenting the important problems in such a framework, along with solutions that appear to be viable at this stage. We have started implementing a prototype of the architecture with encouraging results. These results, we believe, are good initial steps in verifying the viability of our approach.

References

[1] K. Aberer. P-Grid: A Self-organizing Access Structure for P2P Information Systems. In *CoopIS*, 2001.
[2] P. Boncz and C. Treijtel. AmbientDB: Relational Query Processing in a P2P Network. In *DBISP2P*, 2003.
[3] I. Clarke *et al.* Freenet: A Distributed Anonymous Information Storage and Retrieval System. *Lecture Notes in Computer Science*, 2009, 2001.
[4] A. Crainiceanu *et al.* Querying Peer-to-Peer Networks Using P-Trees. In *WebDB*, 2004.
[5] A. Crespo and H. Garcia-Molina. Semantic Overlay Networks for P2P Systems. Technical report, Computer Science Department, Stanford University, 2003.
[6] A. Datta *et al.* Range queries in trie-structured overlays. In *IEEE International Conference on Peer-to-Peer Computing*, 2005.
[7] D. J. DeWitt and J. Gray. Parallel database systems: The future of high performance database systems. *Commun. ACM*, 35(6), 1992.
[8] W. Fan *et al.* Secure XML Querying with Security Views. In *SIGMOD*, 2004.
[9] L. Galanis *et al.* Processing Queries in a Large P2P System. In *CAiSE*, 2003.
[10] P. Ganesan, M. Bawa, and H. Garcia-Molina. Online Balancing of Range-Partitioned Data with Applications to Peer-to-Peer Systems. In *VLDB*, 2004.
[11] A. Halevy *et al.* Piazza: Data management infrastructure for semantic web applications. In *WWW*, 2003.
[12] R. Huebsch *et al.* Querying the Internet with PIER. In *VLDB*, 2003.
[13] G. Karvounarakis *et al.* RQL: A Declarative Query Language for RDF. In *WWW*, 2002.
[14] D. Kossmann. The State of the Art in Distributed Query Processing. *ACM Comp. Surveys*, 32(4):422–469, 2000.
[15] OGSA-DAI. `http://www.ogsadai.org.uk`.
[16] S. Ramabhadran *et al.* Brief announcement: Prefix hash tree. In *PODC*, 2004.
[17] S. Ratnasamy *et al.* A Scalable Content-Addressable Network. In *SIGCOMM*, 2001.
[18] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms*, 2001.
[19] I. Stoica *et al.* Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *SIGCOMM*, 2001.
[20] I. Tatarinov *et al.* The Piazza Peer Data Management Project. *SIGMOD Record*, 32(3), 2003.
[21] P. Valduriez and E. Pacitti. Data Management in Large-scale P2P Systems. In *VECPAR*, 2004.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

GREMO: A GT4 based Resource Monitor

Vijay Sahota and Maozhen Li

School of Engineering and Design
Brunel University, Uxbridge, UB8 3PH
Email: {Vijay.Sahota, Maozhen.Li}@brunel.ac.uk

Abstract

The past few years have seen the Grid rapidly evolving towards a service-oriented computing infrastructure. With the OGSA facilitating this evolution, it is expected that WSRF will be acting as the main an enabling technology to drive the Grid further. Resource monitoring plays a critical role in managing a large-scale Grid system. This paper presents GREMO, a lightweight resource monitor developed with Globus Toolkit 4 (GT4) for monitoring CPU and memory of computing nodes in a Windows and Linux environments.

1. Introduction

The Grid [1] couples a global array of distributed resources that require constant monitoring if any integration and coordination is to take place. By processing this raw monitored data into useful information ensures optimal use of resources, pooling them for large capacity workloads, but still be able to work over a heterogeneous and geographically dispersed environment. Ease of use and accessibility is the major factor for rapid uptake and acceptance of Grid computing, but as usual the commercial aspect in providing services will have the greatest impact, but before the commercial sector can take any interest the Grid must provide a means of guaranteed service. Since monitoring is a key to organising any operations in a computing environment building a history of resource usage, one can perform some intelligent predictions on the state of the network in the near future, hence enabling the Grid to provide a guaranteed service from predicting which services will be available.

To reach as many users as possible with global coverage, the internet provides a universal foundation for communication whilst using existing technology. Its intrinsic property of interoperability is still an issue yet to be resolved in Grid computing. So far the general direction of using Web services [2] has been the main approach, to allow the Grid to operate over the internet whilst enabling utilisation through a standard Web browser, benefiting clients behind firewalls. The past few years have seen the Grid evolving rapidly towards a service-oriented computing infrastructure. The Open Grid Services Architecture (OGSA) [3] has facilitated this evolution. It is expected that Web Services Resource Framework (WSRF) [4] will be acting as an enabling technology to drive this further.

In this paper we present GREMO, a lightweight resource monitor using the Globus Toolkit 4 (GT4) [5], an implementation of WSRF standard, that included the WSN (Web Services notification) specifications [6] to support notifications. Currently, GREMO only monitors CPU and Memory usage, however its design is kept generic so that it can be applied to monitor any Grid resource. Many have tried and succeeded well in producing a monitoring system that works well on a large scale network, optimising programs that use minimal system resources whilst working towards a real time performance, such as Ganglia [7] and Network Weather Service (NWS) [8], but in most cases this entails a complex software set-up, restriction to a certain kind of network/operating system or a lack of the functionality to be as easy to use and accessible as a Web page. GREMO is implemented as a lightweight monitoring system requiring only a standard web server running. Using standard Web Service technologies, it can monitor resources in both Windows and Linux environments.

The rest of the paper is organised as follows: Section 2 briefly reviews WSRF and WSN. Section 3 introduces the design of GREMO, and describes the main components of GREMO. Section 4 presents some experimental results to show the performance of GREMO. Section 5 concludes this paper.

2. WSRF and WSN

WSRF is a set of specifications that specify how to make Web services stateful amongst other aspects. The problem of where to store state involved the introduction for the concept of 'resources' (WS-Resource); a persistent memory for services which may reside in memory, hard disk or even a database. Each resource uses a unique address termed 'endpoint reference' to isolate resources from services enabling other services to use them directly with out having to go through the parent service, given this the introduction of other useful functions were also created.

- WS-Resource Lifetime – manages resources by setting a life time
- WS-Resource Properties (RPs) - many elements to a resource, similar to an object.
- WS-Service Group- enables grouping certain services to aid searching for them.
- WS-Base Faults- returning error exception that may be produced by WS-Resource.
- WS-Addressing- actual address given to services and resources rather than URL, enables one to use resources or Web services independently.

Finally, but key to creating a truly independent running service, the WSN system allows services to independently notify an authority via a SOAP [10] message when changes in a resources occur. Replacing the need for an authority to systematically poll for monitoring data, resulting in the inherent saving in time and bandwidth and with no need for special network conditions. Quite simply having created and invoking the resources like standard Web services, clients can easily be created to modify these RPs in relation to the monitored resource given its qualified name (qName) which is a concatenation of the resources namespace and RP name as a qName type.

An authority can then use their own (client) End Point Reference (ERP) to become a subscriber, and given the qName from which to receive notifications registration can be set, where a listener client will act on received notifications. Once a notification is received, EPR and RP's qName (from the sender) along with its new value can be extracted form the message and then be processed. The addition of this new functionality also means that the WSDL [9] documents also need to show its descriptions of a RP, straying away from the standard format, List 1 shows the additional code need for the WSRF standard.

```
<portType name="RegPortType"
   wsdlpp:extends="wsrpw:GetResourceProperty
   wsntw:NotificationProducer"
 wsrp:ResourceProperties
          ="tns:RegResourceProperties">

   <operation name="cname">
      <input message="tns:CnameInputMessage"/>
      <output message="tns:CnameOutputMessage"/>
   </operation>

</portType>
```

List 1: WS-Resource property definition in WSDL.

Note that the `wsrp: Resource Properties` attribute of the `portType` element specifies what the service's resource properties are. The resource properties must be declared as a type where the monitoring state information is kept. Firstly the 'wsdlpp:extends' attribute allows the use of predefined port types, in this example we have used both the 'get resource' & 'send notification' with bindings automatically created by GT4, so there is no need to specify in the WSDL code.

3. GREMO Architecture

Figure 1 shows GREMO architecture the following sections describe each GREMO components.

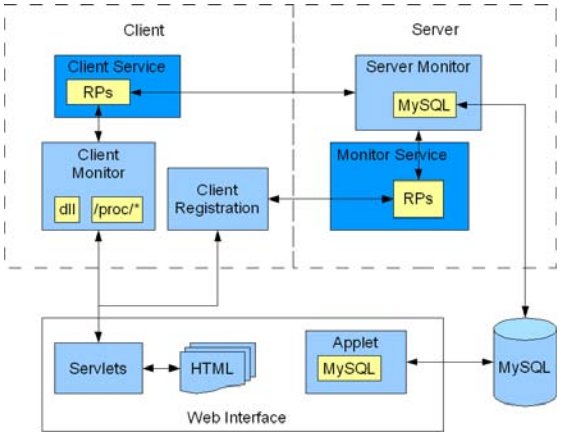


Figure 1: GREMO architecture.

3.1 Client Service

This service runs locally, its main function is to provide RPs that represents the local resources being monitored along with methods to modify them. The notification system is used to alert the subscriber of any changes in these RPs. In this case the service here only has to send out notifications (requiring a web container). Each client with its own service means they have their own set of RPs. This approach means that there is not a single service (server) having to be constantly updated for each user, giving rise to unnecessary instance and processing problems on the receiving side, rather having the client(s) send the notifications. The server simply processes notifications to reduce strain and

improve flow by not having to call and wait for a result/ response, not to mention the delay incurred when a client gets disconnected abruptly.

3.2 Server Service

Similar to the client service, the Server Service provides RPs that represent registration information of clients, with notifications sent when a new user registers. Every time the service is invoked a new set of RPs are created (service instances). Each client has its own set of RPs differing from the Client Service, here one main service which processes all the registration information. Since registration is less periodic than the actual monitoring, this seem the most economic way to create this registry service.

3.3 Client Side Monitoring

Client registration must take place first, given the EPR of the Server Service. The client then invokes the Server Service modifying its RPs allowing the server monitor to use this information to subscribe to the monitoring RPs managed by client. In a similar fashion, the client monitor modifies its Client Service's RPs in accordance to local monitored resources at given set intervals. Both Windows and Linux environments are accommodated to ensure cross platform functionality. Since the information being monitored is CPU and MEM, this information cannot be directly accessed through a Java Virtual Machine (JVM), hence code native to the OS is need. In the case for the Windows part of retrieving monitoring data, pre-complied 'C' dll files were used along with Java Native Interface (JNI) to access them, where as in the case for Linux such values have to be calculated using the /proc/ virtual file system.

3.4 Server Side Monitoring

Here is where the bulk of the monitoring information is processed. The code is a Java application that uses the Server Service RPs as registration information. Subscription to these RPs allow the monitor to subscribe to new users that have registered (monitoring data) as well as adding the registering data to a buffer, constituting of several vectors representing the monitored data including IP addresses, usage values of CPU and memory.

Using the IP address attached on the notification message as a primary key, the monitor modifies its buffers accordingly, whilst adding the values to the mySQL database. In a similarly fashion de-registration follows the same pattern. Fundamental to this tool's functionality is keeping a record of all users & their resources, using their IP addresses as an index for the buffers that are implemented using multiple vectors.

3.5 Storing Monitoring Data

A mySQL table is used as a persistent storage for the monitoring data, as it would be inefficient to keep over 200 values (integer in memory) for each user that can potentially run into the thousands! From keeping the most recent 100 values (each for CPU and memory) a relative history can be produced along with other useful information, but this is only temporary since when a user logs-off their entry is removed from the table.

Used as a buffer for the Server Monitor, information in this table uses IP address as a primary key index. Essential in acting as a temporary buffer is the ability for data to continuously loop around the set 100 fields given for each, monitored type. This means keeping a counter for each monitored resource, checking if the condition has reached 100 (and consequently around wrap back to 1), before executing a mySQL update.

3.6 Web Interface

HTML Web pages are used for both registering and monitoring clients. In this case an HTML form is used as an interface to a Java Servlet which in turn uses the Client registration class, and Client monitoring class to start monitoring, with a DHTML page to update the client on the resources they were monitoring. Using similar code as in the Server side monitor an applet version uses mySQL connector is created allowing a remote administrator to view the current usage of the monitoring service.

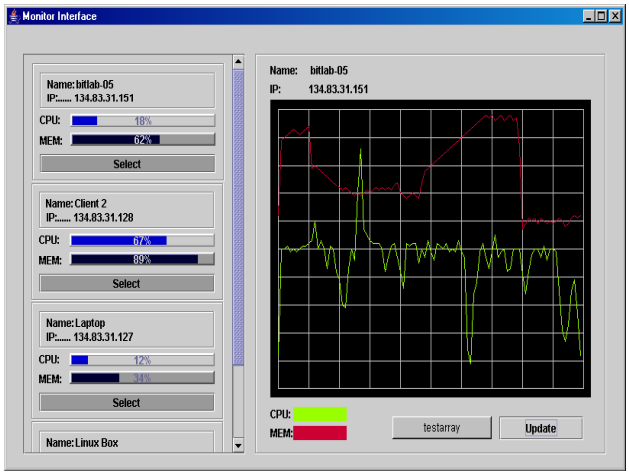


Figure 2: A snapshot of GREMO.

4. GREMO Performance

GREMO is implemented with GT4 on both Windows and Linux platforms. Figure 2 shows a snapshot of GREMO. Ideally GREMO could handle many hundreds of subscribed users, but in the real case there always is usually a limiting factor. GREMO has to processing a large number of SOAP notification messages. Having done a number of experimental tests, using two Pentium IIII workstations running Windows XP, both with 512Mb of RAM running Globus 4.0.0 container, Apache Tomcat 5.0.28 and mySQL 4.1.15 (server monitor only). Tests were carried out with a client sending multiple notifications in burst of 10-500 with the resulting average delays, lags & processing times recorded shown in figure 3.

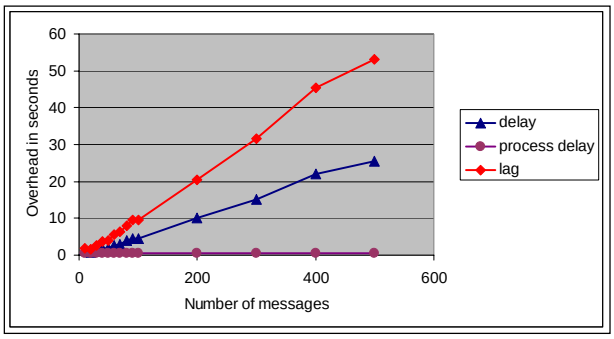


Figure 3: GREMO performance.

Even though testing occurred on a local network, considerable lags and delays can be seen which seem to increase in a linear fashion and start to become quite considerable after burst of 100 messages. Ideally a burst of 40 messages producing a delay and lag of 1.7 and 3.7 seconds respectively seems acceptable. This would be the upper operation limit in this service, making it not feasible to have more than 40 users registered. Note that these delays are for 40 instantaneous whereas 40 users may send 40 messages over a period of time and reduce such delays.

5. Conclusions and Future Work

In this paper we have presented GREMO and have discussed its implementation using the GT4-WSRF notification system. By modifying resource properties of a Web service from which notifications are produced, monitoring data can be logged in a grid environment. This

service offers granularity in that subscription is required and is dynamic, letting the GREMO perform independently without relying on other services to provide this information, as well as keeping track of registered users locally.

Having defined a basic structure the possible uses are widespread; As far as performance goes, we have been using the GT4 notification, which uses Apache Axis. Since at any one time a maximum of two integers are sent, the overhead data wise seems excessive, however, studies have shown that the actual conversion to and from ASCII data is very time consuming leading to performance decreases when the amount of data is increased [11]. Keeping this to a minimal level is beneficiary not only in processing time but also when network traffic is heavy. Time stamping of data to overcome any loss in accuracy history building still has to be implemented.

Taking into account the results shown current implementation of GREMO would be limited to around 40 users. The results also show a steady increase in lag and delay whilst processing delay remain constant suggesting that the GT4 container was processing these notifications at one time, storing the rest in an internal buffer. Work to make this operate in a multiple instance fashion (multi-threaded) will need to be carried out if this is to be a feasible solution. In addition, the building of a large history of resource usage would be the next logical step enabling external services to access this information to aid in performance prediction and job scheduling, producing guaranteed execution times of jobs submitted.

References

[1] I. Foster and C. Kesselman, The Grid, Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers Inc., San Francisco, USA, 1998.

[2] Web services, http://www.w3.org/2002/ws/

[3] Open Grid Services Architecture (OGSA), http://www.globus.org/ogsa/

[4] Web Services Resource Framework (WSRF), http://www.globus.org/wsrf/

[5] Globus toolkit 4 (GT4), http://www.globus.org/toolkit/

[6] Web service notification, http://www-128.ibm.com/developerworks/library/specification/ws-notification/

[7] Ganglia, http://ganglia.sourceforge.net/

[8] Network Weather Service, http://nws.cs.ucsb.edu/

[9] Web Service Description Language (WSDL), http://nws.cs.ucsb.edu/

[10] SOAP, http://www.w3.org/TR/soap/

[11] M. Govindaraju, A. Slominski, K. Chiu, P. Liu, R. van Engelen, M. J. Lewis: Toward Characterizing the Performance of SOAP Toolkits. GRID 2004: 365-372

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Evaluation of Authentication-Authorization Tools for VO security

Jake Wu, Panos Periorellis
School of Computing Science, University of Newcastle,
Newcastle upon Tyne, NE1 7RU, UK
Jake.Wu@newcastle.ac.uk

Abstract

GOLD (Grid-based Information Models to Support the Rapid Innovation of New High Value-Added Chemicals) is concerned with the dynamic formation and management of virtual organizations in order to exploit market opportunities. The project aims to deliver the enabling technology to support the creation, operation and successful dissolution of such virtual organizations. A set of middleware technologies has been designed and being implemented to address issues such as trust, security, contract management and monitoring, information management, etc. for virtual collaboration between companies. In this poster presentation, we will showcase some of the more general requirements for authentication and authorization aspects in GOLD virtual organizations. In conjunction with these requirements we evaluate some of the more popular tools that are currently available in dealing with these issues, together with our own approach that addresses these problems.

1 Introduction

Authentication and authorization mechanisms are integral to the operation of any virtual organization (VO). The GOLD project spent a considerable amount of time and effort gathering a full set of requirements [1] that address the needs of VOs in terms of such mechanisms. We have identified a need for flexible, interoperable solutions that are capable of dealing with the cross-organizational nature of VOs as well as its dynamics in terms of composition. The tools that we discuss in this poster presentation are open source solutions that are popular amongst the e-Science community. We attempt to evaluate those tools in conjunction with a set of requirements for authentication and authorization in VOs that we discuss in the next section. We will present 4 tools and frameworks and the poster will highlight on a comparative evaluation between these tools.

2 Requirements

Virtual organizations bring together a number of autonomic organizations to assess a market opportunity. The duration of this collaboration can be brief or require a larger life span. It is certain that during the lifetime of a VO, the parties that form it will be required to share resources. Hence access of those resources will require crossing of organizational boundaries. In terms of authentication, participants of a VO are expected to have implemented their own security mechanisms to protect resources within their boundaries by some authentication mechanisms. A participant may require access to several resources scattered across several organizational boundaries. The way we solve this problem of multiple logins across resources is by involving 3rd trusted parties that provide security assertions as and when needed where a security assertion is a signed credential token. It should allow single sign on to be achieved without dictating who the trusted parties should be to the VO participants. Moreover, the privacy of user's personal information is protected by minimizing identity flows in the system via this approach. Another solution is federation, by which parties agree on pre-determined trust relations between service providers and identity providers. Beyond federation we also need to provide flexible protocols that allow participants to validate security assertions using authorities that those participants trust.

Regarding authorization, given the dynamicity of VOs and the sensitivity of sharing information, static rights assignment is not sufficient to capture all the eventualities and also VO participants are not expected to be handed permission that last throughout the VO duration. It is more likely that companies will agree limited access or gradual access to their resources depending on progress. In GOLD we want to be able to enable VO to define conceptual boundaries around projects and tasks so that roles and permissions can be scoped. It must also allow for the dynamic activation and deactivation of permissions and roles based on progress monitoring of projects and tasks based on established contracts. We need to adopt fine grained access control mechanisms. In GOLD, the simple subject-object permissions model on which RBAC is based in not sufficient. We need fine grained permissions for instances of roles as well as instances of objects. In the meantime we want to support the delegation of roles, privileges between different levels of authorities. Also in GOLD, given the wide range of policies for access control within a VO and the fact that no single authority governs these policies validation is needed to make sure that there are no logical inconsistencies prior to the workflow enactment.

3 Tools and Evaluation

In this section of the poster presentation, we introduce 4 tools that target similar issues of authentication and authorization. These tools are PERMIS, OASIS, Shibboleth, XACML. We will present these tools and their features individually in this section. In the subsequent section we will provide a detailed evaluation of these tools in conjunction with our requirements.

3.1 PERMIS – PrivilEge and Role Management Infrastructure Standards Validation

PERMIS was funded by Information Society Initiative in Standardization and developed by Salford University. PERMIS dictates a typical role based access control (RBAC) model and aims to provide a solution for managing user credentials when accessing target resources. It is used in electronic transactions in governments, universities or businesses for solving the "authentication of the personal identity of the parties involved" and the "determination of the roles, status, privileges, or other socio-economic attributes" of the individual, through the use of X.509 attribute certificates (ACs) [2]. ACs are widely used to store users' roles and XML-based authorization policies for access control decisions in PERMIS. They are digitally signed by the issuers and stored in the public repositories. Chadwick & Otenko [3] has summarized basic features of PERMIS including being a mechanism for identifying users, specifying policies to govern the actions users can perform, and making access control decisions based on policy checking.

In short, PERMIS is a RBAC system, which bases all the access control decisions on the roles for users and policies. Roles and policies are respectively stored in X.509 ACs, which are then protected by digital signature and kept in the public repository. In the PERMIS architecture, a user makes an access request via an application gateway. In the application gateway the Access Control Enforcement Function (AEF) unit authenticates the user and asks the Access Control Decision Function (ADF) unit if the user is permitted to perform the requested action on the target service provider. The ADF accesses LDAP (Lightweight Directory Access Protocol) directories to retrieve the policy and role ACs, and then makes a granted or denied decision. PERMIS defines its own policy grammar. The typical components that PERMIS XML policy comprises are defined by Bacon et al [4], Chadwick & Otenko [5] [3].

3.2 OASIS - An Open, role-based, Access control architecture for Secure Interworking Services

OASIS is developed by Opera research group in Cambridge Computer Lab [6] and is a RBAC system for open, interworking services in a distributed environment, with services being grouped into domains for the purpose of management. The aim of OASIS is to provide a standard mechanism for users and services in a distributed environment to interwork securely with access control policies enforced. OASIS system is based on Role Membership Certificates (RMC) issued to the users and Credential Records (CR) stored on the servers. One focus of OASIS is the dynamic role activation. For example, in order for a user to possess a role, there may be a set of role activation conditions that must be satisfied. These conditions may include requirements for prerequisite roles, and any other constraints. Roles can be activated or deactivated dynamically as situations arise. OASIS uses appointment certificates (ACs) for associating privileges persistently with membership of some roles and for handling delegation of rights between users. These certificates can be issued to users from some roles with the particular functions and they can serve as a form of credential to satisfy the role activation conditions for a user to activate one or more other roles. Some other key differences are summarized by Bacon et al [4] between OASIS and other typical RBAC schemes (Sandhu R. et al [7] proposed a number of RBAC models).

3.3 Shibboleth – Federated Identity

Shibboleth is a project developed at Internet2/MACE [9]. It is an identity management (user attributes based) system designed to provide federated identity and aims to meet the needs of the higher education and research communities to share secured online services and access restricted digital content. It focuses on providing a way for a user using a web browser to be authorized to access a target site using information held at the user's security domain. It also aims to allow users to access controlled information securely from anywhere without the need of additional authentication process. "Shibboleth is developing architectures, policy structures, practical technologies and an open source implementation to support inter-institutional sharing of web resources subject to access control" [8]. The design of Shibboleth was based on a few key concepts:

a. Federated Administration.
b. Access Control Based on Attributes.
c. Active Management of Privacy.
d. Standards Based.
e. A Framework for Multiple, Scaleable Trust and Policy Sets (Federations).
f. Has defined a standard set of attributes.

Shibboleth system includes two main components: Identity Provider (IdP) and Service Provider (SP) [8][10][11]. IdP is associated with the origin site. SP is associated with the target site. These two components are usually deployed separately but they work together to provide secure access to web based resource. Shibboleth aims to exchange user attributes securely across domains for authorization purpose and PKI is the foundation to build the predetermined trust

relation between Shibboleth components, i.e. IdPs and SPs, of the federated members. Vullings et al [11] points that the main assumption underlying Shibboleth is that the IdP and the SP trust each other within a federation. Some main features of this federation activated system are summarized by Morgan et al [10]. The Shibboleth architecture will be described in details with diagrams in the poster. Shibboleth uses OpenSAML APIs to standardize message and assertion formats, and bases protocol bindings on SAML.

3.4 XACML – eXtensible Access Control Mark-up Language

XACML is an XML based Web service standard for evaluating security sensitive requests against access control policies. It provides standard XML schema for expressing policies, rules based on those policies and conditions. It also specifies a request/response protocol for sending a request and having it approved.

The XACML specification also defines an architecture for handling the entire lifecycle of a request, from its creation through to its evaluation and response. Several components such as the Policy Enforcement Point (PEP) and Policy Decision Point (PDP) transform requests into the standard format before they are evaluated using a rule combining algorisms [12]. The benefits of using XACML as written in the specification are summarized in SUNXACML[12]. XACML allows fine-grained access control and is based on the assumption that a user's request to perform an action on a resource under certain conditions needs an "allow" or "deny" decision.

4 Discussion

The requirements section of this poster describes how virtual organization dynamics can affect decisions regarding the implementation of authentication and authorization mechanisms. In this section, we evaluate the tools based on a few elements we draw from requirements.

4.1 Authentication and SSO

The GOLD system supports SSO via the usage of SAML assertions and related protocols as specified by Liberty Alliance [13], using OpenSAML [14] APIs. PERMIS offers a RBAC authorization infrastructure. However, PERMIS has been built to be authentication agnostic [4]. When a user wishes to access an application controlled by PERMIS, the user must first be authenticated by the application specific AEF. OASIS does not have support for authentication. It is entirely an access control system. Shibboleth implements the SSO mechanism via the use of SAML and the concept of federated identity, so users are only required to sign on once at home organizations. In the GOLD authentication system, we support a variety of tokens, e.g. X.509 digital signature/encryption, SAML assertion. In PERMIS, when authenticating a user the AEF could use digital signatures, Kerberos, or username/password pairs [4]. Shibboleth mainly uses the SAML standard to construct its tokens through OpenSAML [14] APIs which is also developed at Internet2 [9] while other tokens are also supported. The standardization that SAML offers largely facilitates the exchange of security information about users in trust domains and fits nicely with the other WS-* standards utilized in the GOLD system.

4.2 Privacy

Privacy is also an essential issue that a security system needs to address. In the GOLD system, a user is issued with a SAML assertion, or even a GOLD context id in the prototype, and only this assertion or context id flows between the GOLD participants. A user is only authenticated once with his private information without the need to provide them for any additional times. Privacy Protection is not provided in PERMIS. PERMIS has chosen public repositories to store the attribute certificates, which compromises the user's privacy[4]. In OASIS the ACs are not publicly visible when being issued and presented. When ACs go through communication channels they are encrypted under SSL, therefore the privacy is ensured. In Shibboleth, fairly active management of privacy was in place when the system was designed and the users have full controls what personal information is released and to whom.

4.3 Federation and broker style trust

Federation is an indispensable part in the GOLD architecture and offers the GOLD participants freedom of deciding whether they want to trust the identity providers based on a pre-determined trust relationship. PERMIS [2] and OASIS [15] have been developed in a distributed environment. Undoubtedly federation is the paramount issue in the Shibboleth system. However the GOLD system also offers participants a choice of finding alternative independent parties in cases of parties disagreeing on the authorities they trust, without sacrificing any traceability or accountability of credentials. This broker style trust is not currently supported in any of other aforementioned tools.

4.4 Dynamic activation and deactivation of access rights

Central to the authorization requirements is the need for dynamic activation and deactivation of the access rights. As addressed earlier, the GOLD framework supports this by enabling VOs to define conceptual boundaries around projects and tasks so that the roles and permissions can be scoped. Ongoing decisions can be made along with the progresses of projects or tasks in relation to the dynamic activation and deactivation of the access rights. A high level of granularity is provided in this context. These are all achievable by

taking the advantages of standardization and flexibilities that XACML can offer. OASIS [4] provides the means for the dynamic role activation as discussed in the earlier section. In contrast, the role and policy rights assignment in PERMIS are rather persistent and the attribute certificates PERMIS uses are a static representation. According to the PERMIS developers, in the current release of PERMIS, a role is revoked by explicitly deleting the role AC from the LDAP directory using an LDAP browser/admin tool. Shibboleth focuses on attribute-based authorization and it does not mention the dynamic use of policy rights.

4.5 Policy delegation

Also in GOLD, we support the delegation of authorities, where the source of authorities can delegate roles/privileges/rights to the subordinate authorities. X.509 standard, which PERMIS is based on, specifies mechanisms to control the delegation of authority from the source of authority to subordinate attribute authorities. The delegation was not fully supported by the PERMIS implementation in the previous releases. However it is currently supported by the recently developed release according to the developers. A role holder may delegate his/her role to another individual, without the need to have permission to alter the privileges assigned to that individual. Furthermore PERMIS supports role hierarchies. With role hierarchies privileges of subordinate roles can be inherited by superior roles, and a role holder can delegate just a subordinate role instead of the entire role [16]. OASIS uses appointment as introduced earlier, as opposed to the privilege delegation.

5 Conclusion

This poster submission looked at VO requirements in terms of authorization and authentication taking into account the dynamics of a VO, the levels of distrust, the need for federation as well as rights delegation. We have evaluated the 4 tools available under open source licensing which are addressing similar issues. We give practical assessments on what these tools do and how they address issues that we are concerned in conjunction with the requirements we elicited from the project.

References

[1] Periorellis, P., Townson, C. and English, P., 2004 CS-TR: 854 Structural Concepts for Trust, Contract and Security Management for a Virtual Chemical Engineering, School of Computing Science, University of Newcastle.

[2] PERMIS, 2001, Privilege and Role Management Infrastructure Standards Validation, http://www.permis.org/en/index.html

[3] Chadwick D., Otenko S. 2003_1, A comparison of the Akenti and Permis authorization infrastructures, Proceedings of the ITI First International Conference on Information and Communications Technology (ICICT 2003) Cairo University, *pages 5-26*,

[4] Bacon J. et al. 2003, Persistent versus Dynamic Role Membership, 17th IFIP WG3 Annual Working Conference on Data and Application Security, No. 17, pages 344-357

[5] Chadwick D., Otenko A. 2003_2, The PERMIS X.509 role based privilege management infrastructure, Future Generation Computer Systems, Vol. 19, Issue 2, 277-289

[6] OASIS, 2003, An Open, Role-based, Access Control Architecture for Secure Interworking Services, Cambridge University EPSRC project, http://www.cl.cam.ac.uk/Research/SRG/opera/projects/.

[7] Sandhu R., Coyne E., Feinstein H. & Youman C., 1996, Role-Based Access Control Models, IEEE Computer, Volume 29, Number 2.

[8] Shibboleth 2005, Shibboleth Project, Internet2/MACE, http://shibboleth.internet2.edu

[9] Internet2/MACE, 1996-2005, Internet2-Middleware Architecture Committee for Education, http://middleware.internet2.edu/MACE/

[10] Morgan R. L., Cantor S., Carmody S., Hoehn W., & Klingenstein K. 2004, Federated Security: The Shibboleth Approach, Educause Quarterly, Vol. 27, No. 4.

[11] Vullings E., Buchhorn M., Dalziel J. 2005, Secure Federated Access to GRID applications using SAML/XACML.

[12] SUNXACML 2004, Sun's XACML Implementation, http://sunxacml.sourceforge.net

[13] Liberty Alliance Project 2003, Introduction to the Liberty Alliance Identity Architecture, Revision 1.0.

[14] OpenSAML 2005, An Open Source Security Assertion Markup Language implementation" (Internet2) http://www.opensaml.org/

[15] Hine J.H., Yao W., Bacon J., Moody K., 2000, An Architecture for Distributed OASIS Services, Proc. Middleware 2000, Lecture Notes in Computer Science, Vol. 1795, Springer-Verlag, Heidelberg and New York, 107-123.

[16] Chadwick D. & Otento A., 2002, RBAC Policies in XML for X.509 Based Privilege Management, in Security in the Information Society: Visions and Perspectives: IFIP TC11 17th Int. Conf. On Information Security (SEC2002), Cairo, Egypt. Ed. by M. A. Ghonaimy, M. T. El-Hadidi, H.K.Aslan, Kluwer Academic Publishers, pp 39-53

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Simple Grid Access using the Business Process Execution Language*

Clovis Chapman[1], Andrew M. Walker[2], Mark Calleja[3],
Richard P. Bruin[2], Martin T. Dove[2] and Wolfgang Emmerich[1]

[1] Dept. of Computer Science, University College London,
Gower St, London WC1E 6BT, United Kingdom
[2] Dept. of Earth Sciences, University of Cambridge,
Downing Street, Cambridge CB2 3EQ, United Kingdom
[3] Cambridge eScience Centre, Centre for Mathematical Sciences,
University of Cambridge, Wilberforce Road, Cambridge CB3 0EW

Abstract

Scientists require means of exploiting large numbers of grid resources in a fully integrated manner through the definition of computational processes specifying the sequence of tasks and services they require. The deployment of such processes, however, can prove difficult in networked environments, due to the presence of firewalls, software requirements and platform incompatibility. Using the *Business Process Execution Language* (BPEL) standard, we propose here an architecture that builds on a *delegation* model by which scientist may rely on middle-tier services to orchestrate subsets of the processes on their behalf. We define a set of inter-related workflows that correspond to basic patterns observed on the *e*Minerals minigrid. These will enable scientists to incorporate job submission and monitoring, data storage and transfer management, and automated metadata harvesting in a single unified process, which they may control from their desktops using the *Simple Grid Access* tool.

1. Introduction

As grid infrastructures evolve and offer an increasingly important amount of resources and services, it becomes clear that scientists are suffering from physical and software restrictions not addressed by current-generation grid middleware and tools. Scientific processes, such as those defined by the scientists of the *e*Minerals project [1], involve a large number of services and resources, such as job execution services, data stores, visualisation services, etc., and potentially complex interactions between these. The realities of networked heterogeneous environments make the deployment of such processes an extremely difficult task: firewalls, platform incompatibility, software and resource requirements, security, etc. - all these elements can stop the average user from launching and coordinating complex computational processes from their desktops and/or applications of their choice. Ensuring that scientists can retain their current working environments and applications, with minimal or no change, is key to facilitating the transfer to grid environments and enabling the step change in the research that this provides. Existing grid middleware can prove difficult to install, configure and use, and hardly provides the level of integration required to seamlessly incorporate a wide range of services into a unified process.

We have had the opportunity to work on the deployment of several scientific workflows on the *e*Minerals minigrid [2]. While the scientific goals of these processes may differ, they present several similarities and common requirements. Specifically, they require batch job submission and monitoring primitives, the ability to store and retrieve large amounts of produced data and finally the ability to organize and index this data through the definition of corresponding metadata.

The approach that we adopt here is to decompose large scientific processes into a collection of basic patterns that can be fully automated and delegated to third party VO-wide systems, which will orchestrate the execution of these subsets of the process on the user's behalf.

We propose an architecture that builds on this delegation model, relying on the *Business Process Execution Language (BPEL)* [3] and other Web Service tools and standards, such as *GridSAM* and the *Job Submission Description Language (JSDL)*

* Research presented has been funded by NERC through Grant reference numbers NER/T/S/2001/00855, NE/C515698/1 and NE/C515704/1 (*e*Minerals).

[4], to provide VO-wide orchestration and service provision. The increased adoption of Web Services by the Grid community makes the use of this industry-led standard in a Grid environment very appealing: BPEL is an orchestration language, enabling us to build services whose role is to coordinate interactions between Web Services according to specified workflows on a client's behalf.

The architecture relies on the definition and deployment of a number of predefined workflows that correspond to basic patterns observed in our minigrid. Scientists may trigger the execution of these workflows remotely through a *lightweight* self-contained client.

For this purpose, we have implemented the *Simple Grid Access* (SGA) tool: a lightweight, self-contained java-based tool, that enables users to launch job executions and manage submissions from their desktop, including transferring files to and from storage vaults and uploading proxy certificates. SGA can be used directly or incorporated within other applications as an external process, providing users with means of composing more complex workflows at a more abstract level; with applications, tools and languages of his choice – such as simple batch scripting. Alternatively, focus on reusability ensures that users can incorporate our workflows into larger BPEL workflows.

2. Deploying workflows on the eMinerals minigrid

2.1 Scientific workflow requirements

Scientific problems will require the definition of computational processes usually involving the execution of several data management and computational tasks. As an example, we refer to previous work [6], where we tackled the problem of distributing the computations required to study the adsorption of pollutant molecules on a pyrophyllite surface on the e-Minerals minigrid.

The eMinerals is a production level infrastructure encapsulates a wide range of resources across 6 sites in the UK.

It provides essentially 3 categories of services:

- *Compute resources*: consisting of number High Performance Computing (HPC) and High Throughput Computing (HTC) resources. These are typically fronted by Globus 2.4 [5]
- *Data Storage resources:* we rely on the SDSC *Storage Request Broker* (SRB), to provide seamless access to distributed data resources, with a total capacity of 3 terabytes spread across 5 storage vaults.
- *Metadata annotation service (Rcommands):* This service [10] provides means of storing, generating and searching metadata for files stored in SRB vaults to facilitate the indexation and re-use of data.

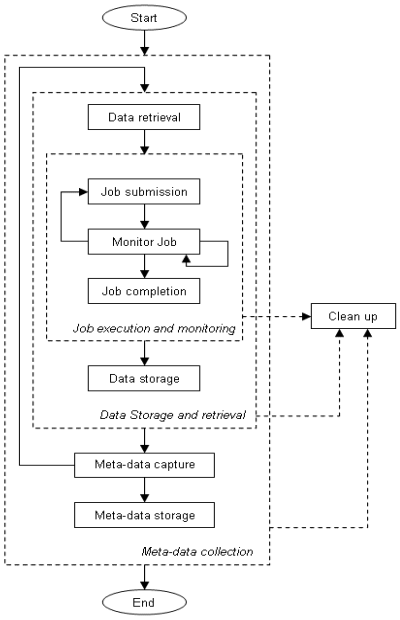


Figure 1: Hierarchical workflow patterns

The number of resources and their geographical distribution clearly highlights the difficulty in deploying and managing complex workflows on such an infrastructure.

The various sites are independently administrated and protected by institutional and/or departmental firewalls. Users also require a certain number of tools, at the very least the Globus toolkit, Condor-G (for job scheduling capabilities), the RCommands front end tools and finally SRB tools for data storage and retrieval. Installation, configuration and maintenance can prove tedious, particularly where administrative access to the machine is not available.

However, a key observation that we have made during our work on workflow deployment is that scientific processes can be decomposed into a number of basic patterns, which take advantage of our services in a unified manner. For each individual job, we require:

- The selection of a target compute resource

- The retrieval and staging of data from our storage resources onto the target resource
- The submission and execution of the job
- The storage of the produced data
- Finally, the automated harvesting of metadata from the produced output, and its storage.

As part of the process, these various stages require fault-handling capabilities and monitoring of state changes. This in itself constitutes a relatively complex but reusable workflow unit (illustrated in figure 1) that can be interleaved with desktop processes, such as retrieving user input for steering purposes, or input and output processing.
The complexity and resource requirements of such a process means that its management and execution is best left to a third party with sufficient resources, for which we rely on the Business Process Execution Language (BPEL).

2.2 The Business Process Execution Language

BPEL aims to enable the orchestration of Web Services, by providing means of composing a collection of Web Service invocations into an executable workflow, itself presented as an *invocable* Web Service.
Much work has been invested in the development of Web Service compliant grid middleware. For example, web service based job submission tools such as GridSAM have been developed, providing a standard job interface to underlying resource management systems such as Condor.
BPEL allows us to specify the process to execute upon request as an XML document. It provides support for controlled flow elements, including sequential or parallel executions, conditional executions, variables, fault handling and XPath and XSLT queries.
Once a workflow has been specified, a BPEL engine will be responsible for orchestrating the workflow on a user's behalf, acting as a middleman between resources and client. Such an approach also provides us with the advantage that the BPEL workflow can be modified independently from the users, considerably easing administration and enabling the transparent addition of new resources and services. Web Services also provide better support for firewalls, by allowing session management over a single port.

3. Implementation

Our implementation is illustrated in figure 2. For job specification, we rely on the Job Submission Description Language (JSDL), an emerging XML based GGF standard [9].

Orchestration Service: At the core of our system we rely on one or more BPEL engines to provide and manage the execution of workflows as specified in section 2.1. We use for this purpose the open source Active BPEL engine to deploy workflows. These workflows, deployed as executable Web Services, will present an interface for clients to submit JSDL documents and corresponding data requirements, and, upon receipt of such a document, will manage the invocations of the various required services.
The Active BPEL engine also provides graphical Web based monitoring tools, providing a detailed view of the execution process, which will ensure that users can check on the progress of their jobs, whilst keeping the client-side tools as light as possible.

Simple Grid Access (SGA) tool: The SGA tool is our self-contained client tool that we have implemented to allow users to specify their job requirements and generate corresponding JSDL and data requirements specifications for submission to a BPEL engine.
We favor here a simple command line interface, which will allow users to specify the various requirements of the job (input files, executables, etc.) as a sequence of arguments, as well as allowing additional attributes (proxy server descriptions etc.) to be obtained from a configuration file, in a predefined location or from local environment settings.
The client also provides additional data staging capabilities. Because client side inbound connections are rarely available, it enables users to upload required files to the SRB vaults before job submission if required through HTTP - as we will explore below, as well as making files to be made available through a variety of means (FTP, HTTP). Files can also be returned to the user's desktop upon job completion. Finally, it also allows users to generate a certificate proxy, which will be uploaded to a credential management service (i.e. myProxy [13]) of the user's choice. Once data and proxy requirements have been handled, the client will invoke the orchestration services to orchestrate the execution of the workflow.
The client has been implemented in Java, ensuring that it will be compatible with most platforms, using the CoG kit [11], Apache Axis [12], Apache HttpClient and other libraries.

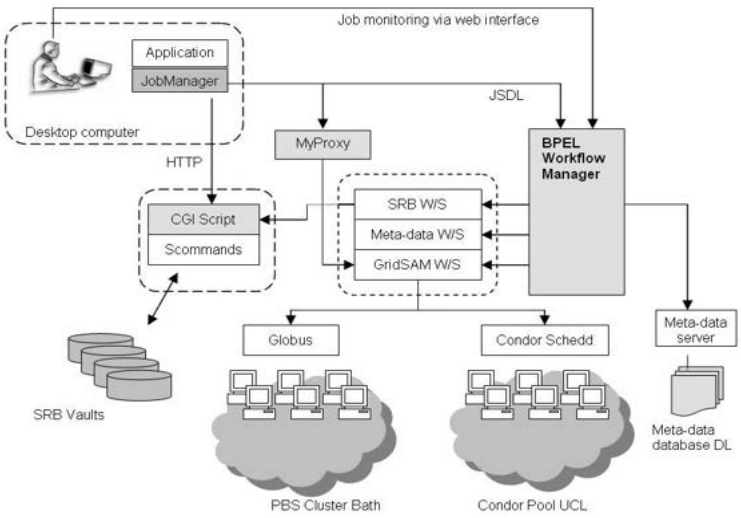


Figure 2: Implementation

Meta-scheduling: We rely here on previous work in which we created a plug-in for the GridSAM job submission service to support submissions to multiple Condor pools, or Globus managed clusters through Condor's Web Service interface [8], effectively transforming GridSAM into a complete metascheduling service. GridSAM also provides support for JSDL, and myProxy, facilitating its integration into the system.

Data Transfer and services: While Web Services are used to control the data transfers, the actual transfer of data is handled through HTTP, in order to avoid the overhead of SOAP based invocations.
For this purpose, we have written a front end wrapper over the SRB tools using CGI that facilitates interaction with the SRB data vaults through HTTP.
Alongside this, we have created a Web Service that will facilitate the remote coordination of transfers between third parties and can be deployed on target resources. The reasons for adopting HTTP are obvious: they facilitate the upload of files for clients in the presence of firewalls, and also enable them to access files through their browser.

Metadata collection: The interface provided by the RCommands Web Service provides means of adding, editing and removing new data sets and studies associated with a particular file, and its physical location (on the SRB).
In addition to basic submission details, detailed information about the data can be obtained by parsing the output files, particularly in XML or CML [7]. We have created an additional metadata collection Web Service that will be responsible for identifying elements of interest given an appropriate RDF file and a suitably rich ontology, relying on the AgentX framework [10].

4. Conclusion

We have begun the process of incorporating the SGA tool into existing workflows and tools used by our scientists.
In particular, we have incorporated SGA into GDIS (GTK Display Interface for Structures), an open source program that allows scientists to visualise molecules, crystals and crystal defects in three dimensions and at the atomic scale [14]. GDIS can act as an interface to create code input and run the simulations on the local machine. By enabling GDIS to invoke our SGA tool, we can now launch calculations onto our minigrid, and take advantage of all our services, with little change to the original code base.

References

1. Dove, M. et al., Environment from the molecular level: an escience testbed project. *Proc. of All Hands Meeting, Nottingham, 2003*
2. Blanshard, L. et al., Grid tool integration within the eMinerals project, in *Proc. Of the All Hands Meeting* Nottingham, 2004.
3. Andrews, T. et al., Business Process Execution Language for Web Services Version 1.1 OASIS, 2003. *http://ifr.sap.com/bpel4ws.*
4. The GridSAM project. *http://www.lesc.ic.ac.uk/gridsam/*
5. The Globus Project. *http://www.globus.org/*
6. White, T. et al., *e*Science methods for the combinatorial chemistry problem of adsorption of pollutant organic molecules on mineral surfaces, in *Proc. Of the All Hands Meeting,* Nottingham, 2005.
7. Murray-Rust, P. et al., Chemical Markup Language and XML Part I. Basic principles, *J. Chem. Inf. Comp. Sci*., **3**9, 928, 1999.
8. Chapman, C. et al. , Condor Birdbath - web service interface to Condor, in *Proc. of the All Hands Meeting*, Nottingham, 2005.
9. Anjomshoaa, A., et al., Job Submission Description Language Specification v1.0, GFD #: GFD-R-P.056, 2005.
10. Tyer, R. P., et al., Automatic metadata capture and grid computing, in *Proc. of the All Hands Meeting*, Nottingham, 2006.
11. Java Cog Kit http://wiki.cogkit.org/
12. Apache Software Foundation. http://www.apache.org/
13. MyProxy http://grid.ncsa.uiuc.edu/myproxy/
14. Fleming, S., and Rohl, A., GDIS: a visualization program for molecular and periodic systems, *Z. Krist. 200 580*, vol.220 pp.580-584, 2005.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Automatic metadata capture and grid computing

RP Tyer, PA Couch, K Kleese van Dam, IT Todorov
CCLRC, Daresbury Laboratory, Warrington, Cheshire WA4 4AD

RP Bruin, TOH White, AM Walker, KF Austen, **MT Dove**
Department of Earth Sciences, University of Cambridge, Downing Street, Cambridge CB2 3EQ

MO Blanchard
Royal Institution, 21 Albemarle Street, London W1S 4BS

Abstract

We report a pragmatic approach to enable non-intrusive automatic metadata harvesting from grid-enabled simulation calculations. The framework, called RCommands, gives users a set of unix line commands and a web interface to the metadata database. The harvesting relies on the use of XML for output data file representation, and new developments of the `my_condor_submit` tool incorporating AgentX.

Introduction

This paper concerns a new set of tools developed by the *e*Minerals project [1] to facilitate automatic metadata harvesting from molecular-scale simulations. The key design requirements were that the tools should be non-intrusive for users, and pragmatic in design. This work represents a close collaboration between the developers and scientists.

The *e*Minerals project studies environmental processes at a molecular level, using a range of atomistic simulation methods. These are performed on the *e*Minerals minigrid [2], which integrates grid computing with grid data management methods based on the San Diego Storage Resource Broker (SRB). Job submission is via the `my_condor_submit` (MCS) tool [2,3], which also supports data and metadata management.

Grid computing enables large scale combinatorial studies. For example, studies of molecular pollutants on mineral surfaces requires comparing the energies of up to 210 members of a single family of molecules (the polychlorobiphenyls). It is necessary to perform calculations of the energies of each molecule in isolation and in contact with a mineral surface, together with repeat calculations using different levels of the theory within the simulation method. Other examples are where calculations are performed over a wide sweep of one or more input parameters, such as temperature or pressure. In all these cases, metadata is used to document the exact conditions of each simulation to enable scientists to locate simulation outputs easily. This replaces the role of the logbook or README file.

Metadata organisation model

The CCLRC model proposes three tiers within which metadadata are organised. The top level is the *study* level. This level is self explanatory. It is possible to associate named collaborators with this level, enabling collaborators to view and edit the metadata within a study. The next level down is the *dataset* level. This is the most abstract level, and users are free to interpret this level in a variety of ways. The third level is the *data object* level. This is the level that is associated with a specific URL (e.g. an SRB URL). The data object may include the files generated by a simulation run, and/or the outputs from subsequent data analysis. In combinatorial studies, there will be many data objects associated with a single dataset, and different types of calculations within a single study are organised with the dataset level. Examples of our usage of this hierarchy are given in Table 1. Our tools can attach metadata to each of the three levels.

Metadata to capture

Typically metadata associated with the study and dataset levels will be added by hand, with the automated metadata capture to be provided at the data object level (although we provide tools for metadata to be automatically captured at the other two levels as well). We define five types of metadata to capture:

Table 1. Examples of how the study / dataset / data object levels have been used to organise data.

Study	Molecular dynamics simulation of silica under pressure	*Ab initio* study of dioxin molecules on clay surface
Data set	Identified by the sample size and the interatomic potential model	Identified by number of chlorine atoms in the molecule and the specific surface site
Data object	Collection on the SRB containing all input and output files	Collection on the SRB containing all input and output files

Simulation metadata: information such as the user who performed the run, the date, the computer run on etc.

Parameter metadata: values for the parameters that control the simulation run, such as temperature, pressure, potential energy functions, cut-offs, number of time steps etc.

Property metadata: values of various properties calculations in the simulation that would be useful for subsequent metadata searches, such as final or average energy or volume.

Code metadata: information to enable users to determine what produced the simulation data, such as code name, version and compilation options.

Arbitrary metadata: strings to enable later indexing into the data, such as whether a calculation is a test or production run.

The RCommand framework

To facilitate automatic metadata capture, we have developed a set of scriptable unix line commands that can upload metadata to the metadata database (Table 2). The RCommands are a standard three-tier application:

Client: A set of binary tools written in C using the gSOAP library. The motivation for using C was the requirement that the tools be as self-contained as possible so they can easily be executed server side via Globus.

Application Server: Written in Java using Axis as the SOAP engine and JDBC for database access. The code essentially maps from procedural web service calls to SQL appropriate for the underlying database [4].

RDBMS: The backend database is served by a Oracle 10g RAC cluster. Although Oracle is used, there is no requirement for any Oracle specific functionality.

One of the main reasons to use a three tier model is that the backend databases are heavily firewalled and cannot be accessed directly from client machines.

The SOAP messages are sent to the application server via SSL-encrypted HTTP. The application server is authenticated using its certificate while the client requests are authenticated using usernames and passwords.

The use of web service technology allows the network related code to be autogenerated. On the server, the Axis SOAP engine is configured to expose specified methods of certain classes as RPC web services. The client code is generated on the fly by gSOAP from the WSDL file produced by Axis.

Metadata Manager: the web interface to the metadata database

The RCommands were written primarily to provide tools that can be used in scripts, but nevertheless they give scientists a useful interface to the metadata database. However, there are cases when a web interface is better, particularly when requiring a graphical overview that cannot be provided by a unix shell interface. Thus RPT has developed a web interface to the metadata database called the "Metadata Manager". This gives an overview of the study level, from which the user can drill down into the various layers. The user can perform a number of the functions that are provided by the RCommands.

The MDM design uses the JSP Model 2 architecture, which is based on the Model-View-Controller (MVC) pattern. In addition, the Front Controller pattern is also used. The majority of the code in the Model layer is common to both the RCommands and the MDM. Hence, as with the RCommands, the database connectivity is provided using the JDBC libraries.

Collecting metadata: the role of XML in output files

Much of the metadata we collect is harvested from output data files. To facilitate this, we have enabled our key simulation programs to write the main output files in XML, using the Chemical Markup Language [5]. CML specifies a number of XML element types for representing lists of data, including:

`metadataList`: contains general properties of the simulation, such as code version;

`parameterList`: contains parameters for with the simulation;

`propertyList`: contains property values computed by the simulation.

Table 2. The ten RCommand unix line commands.

RCommand	Action
`Rinit`	Starts an RCommand session by setting up session files
`Rpasswd`	Changes the password for access to the metadata database
`Rcreate`	Creates study, dataset and data object levels, associating the lower levels with the level immediate above, adding a name to each level, adding a metadata description and topic association in the case of creating a study, and associating a URI in the case of creating a data object.
`Rannotate`	Adds metadata. In the case of studies or datasets, this enables a metadata description, and in the case of datasets and data objects it also enables metadata name/value pairs. It also enables more topics to be associated with a study.
`Rls`	Lists entities within the metadata database. With no parameters, it lists all studies, and with parameters it will list the entries within a study or dataset level. It can also be used to list all possible collaborators or science topics.
`Rget`	Gives the metadata associated with a given study, dataset or data object. In the case of a study, it can also list associated collaborators and science topics.
`Rrm`	Removes entities or parameters from the metadata database.
`Rchmod`	Add or remove co-investigators from a study.
`Rsearch`	Search the metadata database, tuned to search within different levels and against descriptions, name/value pairs and parameters.
`Rexit`	Ends an RCommand session, cleaning up session files.

It is usual to have more than one of each list, particularly the `propertyList`. These lists correspond to the metadata types described above. An example is given in Figure 1.

Automatic metadata capture within a grid computing environment

As described in the introduction, the *e*Minerals scientists run their simulations using the MCS tool [3]. MCS deals with four types of metadata:

1. An arbitrary text string specified by the user.
2. Environment metadata automatically captured from the submission and execution environment.
3. Metadata extracted from the first `metadataList` and `parameterList` elements described in the previous section.
4. Additional metadata extracted from the XML documents. These specifications take the form of expressions with a syntax similar to XPath expressions. These are parsed by MCS and broken down into a single term used to provide the context of the metadata (such as '`FinalEnergy`') and a series of calls to be made to the AgentX library [6].

MCS uses calls to the AgentX library to query documents for data with a specific context. For example, AgentX could be used to find the total energy of a system calculated during a simulation. The user specified expression might have the form:

```
AgentX = FinalEnergy, output.xml:/
PropertyList[title='rolling
averages']/Property
[dictRef='dl_poly:eng_tot']
```

The term providing the name of the metadata item is '`FinalEnergy`' and the document to be queried is `output.xml`. The string following '`output.xml:`' is parsed by MCS and converted to a series of AgentX library calls. In this example, AgentX is asked to locate all the data sets in `output.xml` that relate to the concept '`property`' and which have the reference '`dl_poly:eng_tot`'. The value of this property is extracted and associated with the term `FinalEnergy`. The RCommands are then used to store this name value pair in the metadata database.

AgentX works with a specification of ways to locate data in documents (such as a CML document) that have a well defined content model. There are two components to the AgentX framework:

1. An ontology that specifies terms relating to concepts of interest in the context of this work. These terms relate to classes of real world entities of interest and to their properties. The ontology is specified using OWL and serialised using RDF/XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<cml xmlns="http://www.xml-cml.org/schema"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<metadataList>
 <metadata name="identifier" content="DL_POLY version 3.06 / March 2006"/>
</metadataList>

<parameterList title="control parameters">
 <parameter title="simulation temperature" name="simulation temperature"
 dictRef="dl_poly:temperature">
 <scalar dataType="xsd:double" units="dl_polyUnits:K"> 50.0 </scalar>
 </parameter>
</parameterList>

<propertyList title="rolling averages">
 <property title="total energy" dictRef="dl_poly:eng_tot">
 <scalar dataType="xsd:double" units="dl_polyUnits:eV_mol.-1"> -2.7360E+04
   </scalar>
 </property>
</propertyList>

</cml>
```

Figure 1. Extracts of a CML output file, showing examples of the `metadataList`, `parameterList` *and* `propertyList` *containers.*

2. The mappings, which are used to relate terms in the ontology to document fragment identifiers. For XML documents, these fragment identifiers are XPointer expressions that may be evaluated to locate data sets and data elements in the documents. Each format is associated with its own set of mappings and serialised using RDF/XML.

AgentX is able to retrieve information from arbitrary XML documents, as long as mappings are provided. Mappings exist for an increasing number of simulation codes.

Post-processing using the RParse tool

Although XML output will capture all information associated with the simulation, it is inevitable that the automatic tools may miss some of the metadata; it is not always obvious at the start of a piece of work what properties are of most interest. We have used the AgentX libraries to develop the *Rparse* tool to retrospectively extract metadata by scanning over the XML files contained within the data objects in a single dataset. Rparse uses the SRB Scommands, the RCommands, and the AgentX library. The user specifies a collection in the SRB, the relevant output files, AgentX query expressions, and the dataset into which the metadata is to be inserted.

Examples of applications

We have used the metadata tools for the following applications:

- collaborative studies of adsorption of molecules onto mineral surfaces;
- parameterisation of computations of PCB molecules [7];
- study of silica glass under pressure [8]
- sharing literature search results, with the data object linking to the on-line publication URL.

We are grateful for funding from NERC (grant reference numbers NER/T/S/2001/00855, NE/C515698/1 and NE/C515704/1).

References

1. MT Dove *et al*. The *e*Minerals project: developing the concept of the virtual organisation to support collaborative work on molecular-scale environmental simulations. *Proceedings of All Hands 2005*, pp 1058–1065, 2005
2. M Calleja *et al*. Collaborative grid infrastructure for molecular simulations: The *e*Minerals minigrid as a prototype integrated compute and data grid. *Mol. Simul*. **31**, 303–313 (2005)
3. RP Bruin *et al*. Job submission to grid computing environments. *Proceedings of All Hands 2006*
4. M Doherty, K Kleese, S Sufi. "Database Cluster for e-Science". *Proceedings of UK e-Science All Hands Meeting 2003*, pp 268–271, 2003
5. TOH White *et al*. Development and Use of CML in the *e*Minerals project. *Proceedings of All Hands 2006*
6. PA Couch *et al*. Towards Data Integration for Computational Chemistry. *Proceedings of All Hands 2005*, pp 426–432 (2005)
7. KF Austen *et al*., Using escience to calibrate our tools: parameterisation of quantum mechanical calculations with grid technologies. *Proceedings of All Hands 2006*
8. MT Dove *et al*. Anatomy of a grid-enabled molecular simulation study: the compressibility of amorphous silica. *Proceedings of All Hands 2006*

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Automating Metadata Extraction: Genre Classification

Yunhyong Kim[1,2] and **Seamus Ross**[1,2,3]

[1] Digital Curation Centre (DCC)
[2] Humanities Advanced Technology Information Institute (HATII), University of Glasgow, Glasgow, UK
[3] Oxford Internet Institute (2005/6), University of Oxford
{y.kim, s.ross}@hatii.arts.gla.ac.uk

Abstract

A problem that frequently arises in the management and integration of scientific data is the lack of context and semantics that would link data encoded in disparate ways. To bridge the discrepancy, it often helps to mine scientific texts to aid the understanding of the database. Mining relevant text can be significantly aided by the availability of descriptive and semantic metadata. The Digital Curation Centre (DCC) has undertaken research to automate the extraction of metadata from documents in PDF([22]). Documents may include scientific journal papers, lab notes or even emails. We suggest genre classification as a first step toward automating metadata extraction. The classification method will be built on looking at the documents from five directions; as an object of specific visual format, a layout of strings with characteristic grammar, an object with stylo-metric signatures, an object with meaning and purpose, and an object linked to previously classified objects and external sources. Some results of experiments in relation to the first two directions are described here; they are meant to be indicative of the promise underlying this multi-faceted approach.

1. Background and Objective

Text mining has received attention in recent years as a means of providing semantics to scientific data. For instance, Bio-Mita ([4]) employs text mining to find associations between terms in biological data. Descriptive, administrative, and technical metadata play a key role in the management of digital collections ([25], [15]). As the DELOS/NSF ([8], [9], [10]) and PREMIS working groups ([23]) noted, when done manually, metadata are expensive to create and maintain. The manual collection of metadata can not keep pace with the number of digital objects that need to be documented. Automatic extraction of metadata would be an invaluable step in the automation of appraisal, selection, and ingest of digital material. ERPANET's Packaged Object Ingest Project ([12]) illustrated that only a limited number of automatic extraction tools for metadata are available and these are mostly geared to extracting technical metadata (e.g. DROID ([20]) and Metadata Extraction Tool ([21])). Although there are efforts to provide tools (e.g. MetadataExtractor from University of Waterloo, Dublin Core Initiative ([11], [7]), Automatic Metadata Generation at the Catholic University of Leuven([1])) for extracting limited descriptive metadata (e.g. title, author and keywords) these often rely on structured documents (e.g. HTML and XML) and their precision and usefulness is constrained. Also, we lack an automated extraction tool for high-level semantic metadata (such as content summary) appropriate for use by digital repositories; most work involving the automatic extraction of genres, subject classification and content summary lie scattered around in information extraction and language processing communities(e.g. [17], [24], [26], [27]). Our research is motivated by an effort to address this problem by integrating the methods available in the area of language processing to create a prototype tool for automatically extracting metadata at different semantic levels.

The initial prototype is intended to extract Genre, Author, Title, Date, Identifier, Pagination, Size, Language, Keywords, Composition (e.g. existence and proportion of images, text and links) and Content Summary. Here we discuss genre classification of documents represented in PDF ([22]) as a first step. The ambiguous nature of the term genre is noted by core studies on genre such as Biber

([3]) and Kessler et al. ([17]). We follow Kessler who refers to genre as "any widely recognised class of texts defined by some common communicative purpose or other functional traits, provided the function is connected to some formal cues or commonalities and that the class is extensible". For instance, a scientific research article is a theoretical argument or communication of results relating to a scientific subject usually published in a journal and often starting with a title, followed by author, abstract, and body of text, finally ending with a bibliography. One important aspect of genre classification is that it is distinct from subject classification which can coincide over many genres (e.g., a mathematical paper on number theory versus a news article on the proof of Fermat's Last Theorem).

By beginning with genre classification it is possible to limit the scope of document forms from which to extract other metadata. By reducing the metadata search space metadata such as author, keywords, identification numbers or references can be predicted to appear in a specific style and region within a single genre. Independent work exists on extraction of keywords, subject and summarisation within specific genre which can be combined with genre classification for metadata extraction across domains (e.g. [2], [13], [14], [26]). Resources available for extracting further metadata varies by genre; for instance, research articles unlike newspaper articles come with a list of citations closely related to the original article leading to better subject classification. Genre classification will facilitate automating the identification, selection, and acquisition of materials in keeping with local collecting policies.

We have opted to consider 60 genres and have discussed this elsewhere [initially in 18]. This list does not represent a complete spectrum of possible genres or necessarily an optimal genre classification; it provides a base line from which to assess what is possible. The classification is extensible. We have also focused our attention on information from genres represented in PDF files. Limiting this research to one file type allowed us to bound the problem space further. We selected PDF because it is widely used, is portable, benefits from a variety of processing tools, is flexible enough to support the inclusion of different types of objects (e.g. images, links), and is used to present a diversity of genre.

In the experiments which follow we worked with a data set of 4000 documents collected via the internet using a randomised PDF-grabber. Currently 570 are labelled with one of the 60 genres and manual labelling of the remainder is in progress. A significant amount of disagreement is apparent in labelling genre even between human labellers; we intend to cross check the labelled data later with the help of other labellers. However, the assumption is that an experiment on data labelled by a single labeller, as long as the criteria for the labelling process are consistent, is sufficient to show that a machine can be trained to label data according a preferred schema, thereby warranting further refinement complying with well-designed classification standards.

2. Classifiers

For the experiments described here we implemented of two classifiers. First, an *Image classifier***,** which depends on features extracted from the PDF document when handled as an image. It converts the first page of the PDF file to a low resolution image expressed as pixel values. This is then sectioned into ten regions for an assessment of the number of non-white pixels. Each region is rated as level 0, 1, 2, 3 with the larger number indicating a higher density of non-white space. The result is statistically modelled using the Maximum Entropy principle with MaxEnt developed by Zhang ([28]). Second we implemented a *Language model classifier***,** which depends on an N-gram model on the level of words. N-gram models look at the possibility of word w(N) coming after a string of words W(1), W(2), ..., w(N-1). A popular model is the case when N=3. This has been modelled by the BOW toolkit ([19]) using the default Naïve Bayes model without a stoplist.

3. Experiment Design

An assumption in the two experiments described here is that PDF documents are one of four categories: Business Report, Minutes, Product/Application Description, Scientific Research Article. This is, of course, a false assumption and limiting the scope in this way changes the meaning of the resulting statistics considerably. However, our contention is that high level performance on a limited data set combined with a suitable means of accurately narrowing down the candidates to be labelled would achieve the end objective.

For the first experiment we took the 70 documents in our labelled data belonging to the above four genres, randomly selected a third of them as training data (27 documents) and the remaining documents as test data (43), trained both classifiers on the selected training data, and examined result. In the second experiment we used the same training and test data. We allocated the genres to two groups each containing two genres: Group I included business reports and minutes while Group II encompassed scientific research articles and product descriptions. Then we trained the image classifier to differentiate between the two groups and used this to label the test data as documents of Group I or II. Concurrently we trained two language model classifiers: Classifier I which distinguishes business reports from minutes and Classifier II which labels documents as scientific research articles or product descriptions. Subsequently we took the test documents labelled Group I and labelled them with Classifier I and those labelled Group II and labelled them with Classifier II. We examined the result.

4. Results and Conclusions

The precision and recall for the two experiments are presented in Tables 1 and 2. Although the performance of the language model classifier shown in Table 1 is already high, this, to a great extent, reflects on the four categories chosen. In fact, when the classifier was extended to include 40 genres, it performed only at an accuracy of about 10%. When a different set was employed which included Periodicals, Thesis, Minutes and Instruction/ Guideline, the language model performs at an accuracy of 60.34% and the image classifier on Group I (Periodicals) and Group II (Thesis, Minutes, Instruction/Guideline) performs at 91.37%. Note also that, since Thesis, Minutes and Instruction/Guidelines can be intuitively predicted to have distinct linguistic characteristics, the language classifier's performance on each group is also predicted to perform at a high level of accuracy (results pending). It is clear from the two examples that such a high performance can not be expected for any collection of genres. Judging from the result of the classifiers, the current situation seems to be a case of four categories which are similar under the image classifier but which differ linguistically. A secondary reason for involving images in the classification and information extraction process arises because some PDF files are textually inaccessible due to password protection, and even when text is extracted, text processing tools are quite strict in their requirements for input data. In this respect, images are much more stable and manageable. Combining a soft decision image classifier with the language model both increases the overall accuracy and results in a notable increase in recall for most of the four genres (see Table 2).

Table 1. Result of Language Model Only

Overall accuracy: 77%		
Genres	Prec (%)	Rec (%)
Business Report	83	50
Sci. Res. Article	88	80
Minutes	64	100
Product Desc	90	90

Table 2. Result of Image & Language Model

Overall accuracy: 87.5%		
Genres	Prec (%)	Rec (%)
Business Report	83	50
Sci. Res. Article	75	90
Minutes	71	100
Product Desc	90	100

The results of the experiments indicate that the combined approach is a promising candidate for further experimentation with a wider range of genres. The experiments show that, although there is a lot of confusion visually and linguistically over all 60 genres, subgroups of the genres exhibit statistically well-behaved characteristics. This encourages the search for groups which are similar or different visually or linguistically. Further experiments are planned to enable us to refine this hypothesis.

Further improvement can be envisioned, including integrating more classifiers. An *extended image classifier* could examine pages other than the just first page (as done here), or examine the image of several pages in combination: different pages may have different levels of impact on genre classification, while processing several pages in combination may provide more information.. A *language model classifier on the level of POS and phrases would use* a N-gram language model built on the Part-of-speech tags (for instance, tags denoting words as a verb, noun or preposition) of the underlying text of the document and also on partial chunks resulting from detection of phrases (e.g. noun, verb or prepositional phrases). A *stylometric classifier* taking its cue from positioning of text and image blocks, font

styles, font size, length of the document, average sentence lengths and word lengths. A *semantic classifier* would combine extraction of keywords, subjective or objective noun phrases (e.g. using [24]). Finally a c*lassifier based on available external information* such features as name of the journal, subject or address of the webpage and anchor texts can be gathered for statistical analysis or rule-based classification

5. Putting it into Context

The genre extractor provides the basis for constructing an efficient tool. Extension of the tool to extract author, title, date, identifier, keywords, language, summarizations, and other compositional properties can be targeted based upon genre and will, thereby, improve the precision of these other extractors. When the genre classifier is refined for PDF documents, extending it to cover other document format types (e.g. Open Office, Word, LATEX) will be straightforward. Our aim is eventually to pass the prototype to colleagues in the Digital Curation Centre ([6]) who will integrate it with other extraction tools and eventually an automated ingest model.

Acknowledgements

The Digital Curation Centre (DCC) is supported by a grant from the Joint Information Systems Committee (JISC) and the e-Science Core Programme of the Engineering and Physical Sciences Research Council (EPSRC). The EPSRC supports (GR/T07374/01) the DCC's research programme.

References

[1] Automatic Metadata Generation, http://www.cs.kuleuven.ac.be/~hmdb/amg/documentation.php

[2] Bekkerman R, McCallum A, and Huang G, 'Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora', *CIIR Technical Report*, IR-418 (2004).

[3] Biber D, *Dimensions of Register Variation: a Cross-Linguistic Comparison*, Cambridge (1995).

[4]Bio-Mita, http://personalpages.manchester.ac.uk/staff/G.Nenadic/BioMITA.htm

[5]Boese E S, 'Stereotyping the web: genre classification of web documents', Master's thesis, Colorado State University (2005).

[6] Digital Curation Centre, http://www.dcc.ac.uk

[7] DC-dot, Dublin Core metadata editor, http://www.ukoln.ac.uk/metadata/dcdot/

[8] DELOS, http://www.delos.info/

[9] NSF, http:// www.dli2.nsf.gov/intl.html

[10] DELOS/NSF Working Groups, 'Reference Models for Digital Libraries: Actors and Roles' (2003), http://www.dli2.nsf.gov/internationalprojects/working_group_reports/actors_final_report.html

[11] Dublin Core Initiative, http://dublincore.org/tools/#automaticextraction

[12] ERPANET, Packaged Object Ingest Project, http://www.erpanet.org/events/2003/rome/presentations/ ross rusbridge pres.pdf

[13] Giufirida G, Shek E, and Yang J, 'Knowledgebased Metadata Extraction from PostScript File', *Proc. 5th ACM Intl. conf. Digital Libraries* (2000) 77-84.

[14] Han H, Giles L, Manavoglu E, Zha H, Zhang Z and Fox E A, 'Automatic Document Metadata Extraction using Support Vector Machines', *Proc. 3rd ACM/IEEE-CS conf. Digital Libraries* (2000) 37-48.

[15] NSF-DELOS Working Group on Digital Archiving: 'Invest to Save', Report DELOS and NSF Workgroup on Digital Preservation and Archiving (2003) http://eprints.erpanet.org/94/01/NSF_Delos_WG_Pres_final.pdf

[16] Karlgren J and Cutting D, 'Recognizing Text Genres with Simple Metric using Discriminant Analysis', *Proc. 15th conf. Comp. Ling*., Vol 2 (1994) 1071-1075

[17] Kessler B, Nunberg G, Schuetze H, 'Automatic Detection of Text Genre', *Proc. 35th Ann. Meeting ACL* (1997) 32-38.

[18] Kim Y and Ross S, Genre Classification in Automated Ingest and Appraisal Metadata, J. Gonzalo et al. (Eds.): *ECDL 2006*, LNCS 4172, 63–74, 2006.

[19] McCallum A, Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering, http://www.cs.cmu.edu/ mccallum/bow/ (1998)

[20] National Archives, DROID (Digital Object Identification), http://www.nationalarchives.gov.uk/aboutapps/pronom/droid.htm

[21] National Library of New Zealand, Metadata Extraction Tool, http://www.natlib.govt.nz/en/whatsnew/4initiatives.html#extraction

[22] Adobe Acrobat PDF specification, http://partners.adobe.com/public/developer/pdf/index_reference.html

[23] PREMIS Working Group, http://www.oclc.org/research/projects/pmwg/

[24] Riloff E, Wiebe J, and Wilson T, `Learning Subjective Nouns using Extraction Pattern Bootstrapping', *Proc. 7th CoNLL*, (2003) 25-32

[25] Ross S and Hedstrom M, 'Preservation Research and Sustainable Digital Libraries', *Int Journal of Digital Libraries* (Springer) (2005) DOI: 10.1007/s00799-004-0099-3.

[26] Sebastiani F, 'Machine Learning in Automated Text Categorization', *ACM Computing Surveys*, Vol. 34 (2002) 1-47.

[27] Witte R, Krestel R, and Bergler S, 'ERSS 2005:Coreference-based Summarization Reloaded', *DUC 2005 Document Understanding Workshop*.

[28] Zhang L, Maximum Entropy Toolkit for Python and C++, LGPL license, http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Solving Grid interoperability between 2nd and 3rd generation Grids by the integrated P-GRADE/GEMLCA portal

Tamas Kiss[1], Peter Kacsuk[1,2], Gabor Terstyanszky[1], Thierry Delaitre[1], Gabor Kecskemeti[2], Stephen Winter[1]

[1]Centre for Parallel Computing, University of Westminster
115 New Cavendish Street, London, W1W 6UW
e-mail: kisst@wmin.ac.uk

[2]MTA SZTAKI, 1111 Kende utca 13
Budapest, Hungary

Abstract

Grid interoperability has recently become a major issue at Grid forums. Most of the current ideas try to solve the problem at the middleware level where unfortunately too many components (information system, broker, etc.) should be made interoperable. As an alternative concept the P-GRADE/GEMLCA portal is the first Grid portal that targets the problem at the level of workflows. Different components of a workflow can be executed simultaneously in several Grids providing access to a larger set of resources and enabling the user to exploit more parallelism than inside one Grid. The paper describes how the P-GRADE Portal and GEMLCA enable the execution of workflows in multiple Grids based on different 2nd (GT2 and LCG-2) and 3rd (GT4 and g-Lite) generation Grid middleware.

1. Introduction

There have been several attempts to make existing production Grids and Grid technologies interoperable. A well-known example is the work carried out in the framework of the GRIP European project to make Globus and Unicore interoperable [1]. Recently a new EU project called as the OMII Europe has been launched to solve interoperability between GT4 [2], gLite [3] and Unicore [4] at several levels including job submission, security and portal levels. However, the portal level interoperability only means that Gridsphere [5] is going to be ported for all the mentioned Grid middleware. Other examples of trying to solve interoperability at the job submission level include the new Condor version [6] that is able to submit jobs to different GT versions, Unicore and NorduGrid [7], or GridSAM [17] that aims to provide a Web Service for submitting and monitoring jobs managed by a variety of Distributed Resource Managers.

In the current paper we show that interoperability can be solved in a much higher level, namely at the workflow level that could be part of a Grid portal. Indeed, P-GRADE Grid portal [8] is the first Grid portal that tries to solve the interoperability problem at the workflow level with great success. It means that the components of a workflow can be executed simultaneously in several Grids. In this way the user can exploit more parallelism than inside one Grid. More than that the workflow-level completely hides the low-level Grid details for the end-user who has not to learn the low level Grid commands of different Grids.

2. Connecting Grid Generations and Technologies with the P-GRADE Portal and GEMLCA

Here we present experiments and demonstrations that were specifically designed to illustrate how to make different Grid solutions interoperable at the workflow-level. System administrators of existing Grids can immediately utilise these solutions in order to extend the capabilities of their infrastructure without compromising its current reliable operation.

2.1 Connecting Second Generation Grids

Most of the current production Grid systems are based on second generation Grid technology. The basis of the underlying middleware is in most cases the Globus Toolkit 2, however because of

substantial variations and modifications, these Grids are not naturally interoperable with each other. As the P-GRADE portal supports access to multiple Grids at the same time, and as it also supports both LCG and Globus-based Grids, the portal can be utilised to connect these Grid systems and map the execution of different workflow components to different resources in different Grids. As the portal is also integrated with the GEMLCA legacy code repository (GEMLCA-R) [18], users can not only submit their own executable to Grid resources but can also browse the repository and select executables published by other users.

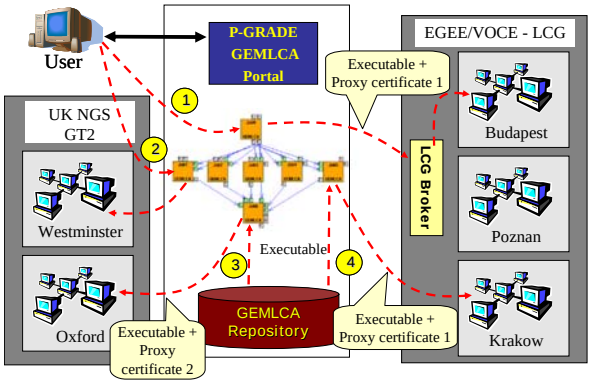


Figure 1. Connecting GT2 and LCG-based Grids

The experiment presented on Figure 1 illustrates how different jobs of a workflow can be mapped to resources within the LCG-based EGEE production Grid [11] and the GT2-based UK NGS [10]. If a user has access to both of these Grids, using the same or different certificates then jobs of a workflow can be mapped in different ways. The user can submit his executable to any of the Grids (1 and 2). In case of LCG-based Grids the LCG broker can also be used besides direct mapping. Executables can also be selected from the GEMLCA repository and submitted (3 and 4), currently only by direct mapping, to any Grid sites. Any P-GRADE GEMLCA portal installation is capable to support the above functionalities. Portal administrators only have to define both Grids with their default resources, and users have to assign the appropriate certificate to each Grid before submitting workflows. As there are currently many users in Europe with access rights to both of these Grids, they can easily utilise the combined power of the resources.

2.2 Extending Second Generation Grids with Third Generation Resources

Most of the production Grids [10] [11] [13] [14] [15] are based on second generation middleware at the moment. However, most of them are considering the transition to service oriented middleware, like GT4 or gLite. The P-GRADE portal, extended with GEMLCA (Grid Execution Management for Legacy Code Applications) [9], is capable to seamlessly assist this transition. GT4 GEMLCA resources can be added to GT2- or LCG-based Grids without any modification of the underlying infrastructure. In this case the GT4 resource is becoming an integral part of the original Grid. Users can map workflow components to these resources using the proxy certificate accepted by that particular Grid, without being aware of the differences of the underlying layers. Figure 2 illustrates, that a GT4 GEMLCA resource, set up at University of Westminster, is added at workflow level to the GT2-based UK NGS. Users can still utilise GT2-based job submission either directly (1) or from GEMLCA-R (2). However, they can also invoke GEMLCA legacy code services (3) through SOAP-XML service invocation but in a user transparent way. This GT4 GEMLCA resource has already been working at production level and available for every NGS user since February 2006.

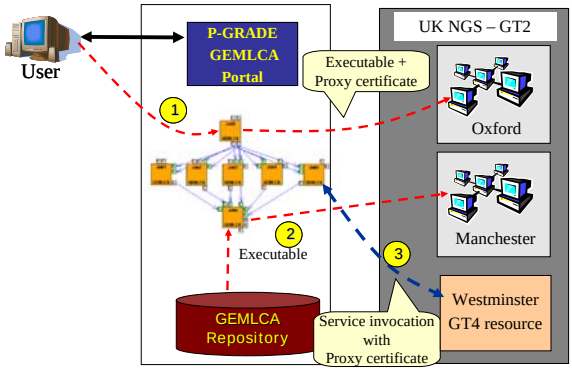


Figure 2. Extending GT2 Grids with GT4 resources

2.3 Connecting Second and Third Generation Grids

As the P-GRADE portal is capable to connect to multiple Grids, and as through GEMLCA it also supports service-oriented Grid systems, it became possible to connect separate GT2- and GT4-based Grids from the same portal at

workflow-level. In order to demonstrate this concept the UK NGS was connected to a GT4 testbed, called the Westfocus Grid [12], comprising clusters at Westminster and Brunel Universities. The GT2 and GT4 Grids were defined as separate Grids in separate administrative domains that could potentially accept different user certificates. As it is illustrated on Figure 3 jobs within a workflow can be submitted to GT2 resources within the NGS (1 and 2), or could be services deployed in the Westfocus Grid (3 and 4).

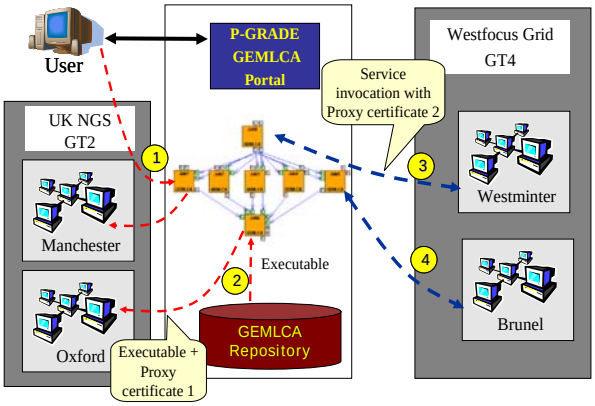


Figure 3. Connecting GT2 and GT4 Grids

2.4 Traffic Simulation Workflow Bridging Grid Generations and Middleware

In order to demonstrate the described capabilities of the P-GRADE GEMLCA portal, a workflow analysing urban car traffic was created [16] (see Fig. 4). The workflow consists of three different applications. Manhattan (job0) is a road network generator that generates input for traffic simulators, like MadCiy. MadCity (jobs 1 and 2) is a discrete-time traffic simulator that simulates traffic on a road network and indicates how the traffic density is changing on different roads and junctions. Finally, a comparator component (job 3) inputs the results of several simulations and creates a graph showing how the traffic density in time depends on several input parameters. In the presented example, jobs of the workflow were mapped to 3 different Grids based on three different underlying middleware. The Manhattan generator (job0) was running in Poznan on the LCG-based EGEE Grid submitted directly there as a job. The first simulator (job1) was a GEMLCA GT4 legacy code service that has been deployed at Westminster University within the Westfocus Grid. The second simulator (job2) was a legacy code submitted from the GEMLCA repository to the GT2 NGS site at Manchester. Finally, the comparator component was a GT2 job submitted directly to Oxford within the NGS. Besides the different underlying architectures the Grids were also using different proxy certificates. The execution graph in figure 5 illustrates that the workflow completed successfully demonstrating the workflow level interoperability of this very diverse infrastructure.

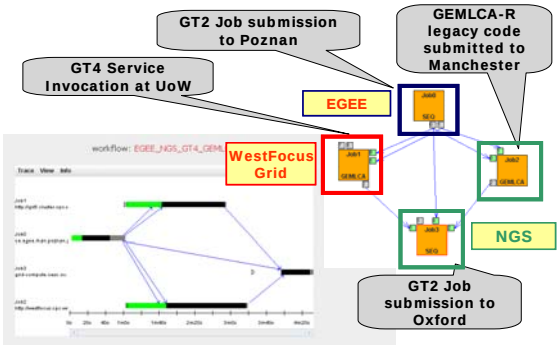


Figure 4. Traffic Simulation Workflow Running in 3 Different Grids

3. Conclusion and Further Work

We have shown in the paper that the P-GRADE portal extended with GEMLCA technology can serve as a bridge between different production Grids that are based on different Grid middleware technologies. The obvious advantages of such a bridge are the following:

1. end users can access any of these Grids from their workflow,
2. virtualization, resource sharing and collaboration can be realized through the boundaries of different production Grids,
3. porting the applications between production Grids does not require any porting efforts.

The P-GRADE portal already serves the SEEGRID (South-East European Grid) [13], the VOCE (Central European VO of EGEE), the HunGrid (the Hungarian VO of EGEE) and the EGRID (the economics Grid VO of EGEE) infrastructures. The P-GRADE GEMLCA portal is available as a service for the UK NGS (GT2) and for the Westfocus Grid (GT4).

Future work, on one hand aims to extend the existing EGEE P-GRADE portals with the GEMLCA service and to connect the P-GRADE/GEMLCA portal to some other large production Grids, namely the NorduGrid and the Open Science Grid.

Work is also carried out in order to extend the service support capabilities of the portal. The GEMLCA – P-GRADE portal is already capable to support GT4-based Grids. However, this support is limited to legacy codes that are presented as Grid services through GEMLCA. The aim is to incorporate native GT4 services and pure Web services into the portal with minimal modification of the current architecture. In order to achieve this Web and GT4 Grid services are handled by modified GEMLCA resources. These GEMLCA resources are used to deploy the Web/Grid service by generating a GEMLCA specific interface description file from the WSDL automatically, to browse the GEMLCA repository, select the required service and set its parameter values, and finally to invoke the service at workflow execution time.

References

[1] Michael Rambadt, Philipp Wieder: UNICORE - Globus: Interoperability of Grid Infrastructures, Conf. Proc of the Cray User Group Summit 2002, Manchester, UK.

[2] Globus Team, Globus Toolkit 4.0 Release Manuals, http://www.globus.org/toolkit/docs/4.0/

[3] The gLite website, http://glite.web.cern.ch/glite/

[4] The Unicore website in wikipedia, http://en.wikipedia.org/wiki/UNICORE

[5] J. Novotny, M. Russell, O. Wehrens: GridSphere, "An Advanced Portal Framework", Conf. Proc. of the 30th EUROMICRO Conference, August 31st - September 3rd 2004, Rennes, France.

[6] Condor Team, University of Wisconsin-Madison: Condor Version 6.4.7 Manual, http://www.cs.wisc.edu/condor/manual/v6.4/

[7] The NorduGrid web page, http://www.nordugrid.org/

[8] P. Kacsuk and G. Sipos: Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal, Journal of Grid Computing Vol. 3. No. 3-4., 2005, Springer, 1570-7873, pp 221-238

[9] T. Delaittre, T. Kiss, A. Goyeneche, G. Terstyanszky, S.Winter, P. Kacsuk: GEMLCA: Running Legacy Code Applications as Grid Services, Journal of Grid Computing Vol. 3. No. 1-2., 2005, Springer, 1570-7873, pp 75-90

[10] The UK National Grid Service Website, http://www.ngs.ac.uk/

[11] The EGEE web page, http://public.eu-egee.org/

[12] The WestFocus Gridalliance web page, http://www.gridalliance.co.uk/

[13] The SEE-GRID web page, http://www.see-grid.org/

[14] The Open Science Grid Website, http://www.opensciencegrid.org/

[15] The TeraGrid Website, http://www.teragrid.org

[16] T. Delaitre, A.Goyeneche, T.Kiss, G.Z. Terstyanszky, N. Weingarten, P. Maselino, A. Gourgoulis, S.C. Winter: Traffic Simulation in P-Grade as a Grid Service, Conf. Proc. of the DAPSYS 2004 Conference, pp 129-136, ISBN 0-387-23094-7, September 19-22, 2004, Budapest, Hungary

[17] William Lee, A. Stephen McGough, and John Darlington: Performance Evaluation of the GridSAM Job Submission and Monitoring System, Conf. Proc. of the UK e-Science All-hands meeting, 2005, http://www.allhands.org.uk/2005/proceedings/papers/541.pdf

[18] T. Kiss, G. Terstyanszky, G. Kecskemeti, Sz. Illes, T. Delaittre, S. Winter, P. Kacsuk, G. Sipos : Legacy Code Support for Production Grids, Conf. Proc. of the Grid 2005 - 6th IEEE/ACM International Workshop on Grid Computing November 13-14, 2005, Seattle, Washington, USA

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

The Combechem MQTT LEGO Microscope

A grid enabled scientific apparatus demonstrator

J. M. Robinson[a], J. G. Frey[a], D. C. DeRoure[b], A. J. Stanford-Clark[c], A. D. Reynolds[c], B. V. Bedi[c], D. Conway-Jones[c]
Email Contact : j.m.robinson@soton.ac.uk

[a]University of Southampton, School of Chemistry, Highfield, Southampton, SO17 1BJ, United Kingdom
[b]University of Southampton, School of Electronics and Computer Science, Highfield, Southampton, SO17 1BJ, United Kingdom
[c]IBM UK Laboratories, Hursley Park, Winchester, SO21 2JN, United Kingdom

Abstract

Grid computing impacts directly on the experimental scientific laboratory in the areas of monitoring and remote control of experiments, and the storage, processing and dissemination of the resulting data. We highlight some of the issues in extending the use of an MQ Telemetry Transport (MQTT) broker from facilitating the remote monitoring of an experiment and its environment to the remote control of an apparatus. To demonstrate these techniques, an Intel-Play QX3 microscope has been "grid-enabled" using a combination of software to control the microscope imaging, and sample handling hardware built from LEGO Mindstorms. The whole system is controlled remotely by passing messages using an IBM WebSphere Message Broker.

1 Background

As computer control becomes increasingly pervasive in the laboratory, it becomes easier to take advantage of automated and semi-automated experimental procedures to enhance the safety, throughput, and quality of data .

An absent exprimenter's expertise can be applied retrospectively and to future experiments but not immediately, unless they have the the ability to monitor the experiment remotely (1). By adding remote control to this monitoring, the experimenter can then steer and modify the experiment without needing physical access. Whilst being of use to the local experimenter this technology opens the possibility of remote, realtime, collaboration with researchers and experts from beyond the local campus. In previous work the use of middleware to support a Publish-Subscribe (or Pub-Sub) methodology(2) for experiment monitoring has been investigated(3). MQTT(4) messages were passed by an IBM MicroBroker to relay information on the laboratory environment to remote users. The same technology can be used to pass messages from the remote user to the laboratory.

1.1 Safety and Physical Security

Remote monitoring and control technologies such as MQTT and the message broker can be used to implement "lights-out" operations of equipment. Though this can be problematic for laboratory personel if the equipment or process is inherently dangerous. The use of local mechanical and electrical interlocks can help to avoid unintended local consequences of a remote action, and by adding remote monitoring the operator is placed in a more knowledgeable position. Security devices such as PIR sensors and door/pressure switches can be used to raise an alert if anyone unexpectedly enters a room or building. Identification technology such as RFID tagging can also be used to verify the identity of personnel who enter or leave a controlled area, using the same MQTT message channel to alert the appropriate parties. Similarly local operators can monitor the status of remote controlers by similar means.

2 Implementation

2.1 Building the MQTT LEGO Microscope

The project can be broken down into three discrete sections, Physical Construction (the LEGO bricks), RCX Software (the link between MQTT and the LEGO sensors and Motors), and Microscope Software (controlling the camera hardware).

LEGO Construction

The LEGO construction around the microscope needs to support the microscope and sample, to provide sample selection/movement and focusing ability. The initial design involved focusing the sample by translating the microscope vertically. The weight of the microscope, and quantity of available LEGO dictatedate that is was more approriate to move the sample vertically below the microscope, keeping the microscope static. The sample platform was driven via a rack and pinion lift powered by a geared down Mindstorms motor. A combination of gearing and driving the motor for very short time periods, allowed the fine control needed to bring the sample into focus even at the highest magnification settings. For protection the sample carriage was equipped with vertical limit switches. Both limits (top and bottom of movement) were connected to the same sensor input (wired in parallel) and the limit reached inferred from the current direction of travel. In a

further experiment the microsope was mounted on a LEGO tractor which could be driven over a floor mounted sample.

RCX Software

The LEGO RCX drive software is split into two parts. Both of these parts originate from IBM Hursley and were modified for this project. The LEGO IR tower and the IR receiver on the RCX are used to link the RCX to the MQTT network in real-time however due to the limited processing capabilities of the RCX a full MQTT protocol stack isn't used. The implementation used here uses single integers to trigger commands on the RCX and a two integer pair to receive a topic and message from the RCX for publishing over MQTT.

This solution requires two bits of code. One that runs on the RCX, and one running on a host PC, which bridges between the messages on IR link and MQTT messages over a TCP/IP network. To allow more extensive programming of the RCX, the standard LEGO firmware is removed and the "lejos" system installed instead(5). Lejos provides a basic Java based programming environment on the RCX and PC side libraries to enable communication with the RCX over the IR link. This makes programming interactions between MQTT and the RCX easier as the standard MQTT Java libraries can be used. The PC based bridge subscribes to a topic from an MQTT broker, and on receiving messages retransmits the payload over the IR link (where the message payload is assumed to only contain an integer). The software on the RCX receives messages over the IR link from the PC. If the integer received corresponds to one of the expected controls codes, the appropriate effector is triggered, otherwise the message is ignored. On receiving a sensor event (the lejos code allows event triggers to be tied to changes in sensor state) the RCX may perform simple processing on the data, triggering effectors as required, and then transmits a message over the IR link to the PC. This is shown schematically in Figure 1.

In the case of the LEGO Microscope there is presently one motor attached to motor control A which drives the focus position and two limit switches (one for each end of travel) connected in parallel to sensor input 1. The RCX accepts codes to stop the motor, and to run it forwards and backwards. It also accepts commands to "bump" the motor in either direction - this drives the motor for 50ms in the direction requested, then stops it. When the limit switches are triggered, the code firstly stops the motors, then backs the motor off in the opposite direction for 200 ms (to release the limit condition), it then publishes a message over the IR link stating which limit switch has been triggered.

Figure 2 shows this program flow schematically.

Microscope Software

The IntelPlay Microscope is viewed from the computer as if it were a webcam. It uses a cpia chipset(6), which is driven by the cpia.ko Linux kernel module. The module writer made control of the camera internal processing easy through the use of the Linux proc interface(7). Through this interface we are able to read and control Sample Lighting, and Image Brightness, Colour Saturation and Hue.

In the current system images are transmitted from the microscope using a simple webcam package(8). This presents images as either JPEG static frames, or as a multipart JPEG, moving them over a HTTP connection. It is planned to investigate other transmission methods, including sending the images over the MQTT broker, at a later stage. Control of the cameras internal processing and sample illuminators is provided by a Linux proc interface to the camera driver. This interface is presented as a file within the proc file system on the Linux machine controlling the microscope. Commands of the form "Toplight:on" are written into this file to control any of the parameters mentioned above. A Java MQTT client subscribes to a microscope control topic on the broker, and on receiving control messages writes the appropriate commands into the camera's control interface.

2.2 Frontend Control.

By using MQTT to pass messages within a remote control system frontend interfaces can be developed using a range of different programming languages, on a variety of platforms. For this demonstrator, the remote control interface was presented as a webpage. Within this page the user is presented with apparatus controls using HTML form elements alongside images from the microscopes camera.

The use of a webpage also provided the benefit that the web browser dealt with transfer and display of the images. The webcam server provided static JPEG images and a javascript timer can then be used to periodically refresh the image.

MQTT control messages are generated using a Perl CGI script running on the webserver. When the end user triggers one of the control buttons on the webform, the CGI script is called, which connects to the broker and publishes messages as expected by the RCX or microscope software.

3 Discussion

The use of LEGO components allows for the rapid development of mechanical solutions that may take weeks to produce using traditional manufacturing processes. The use of LEGO Mindstorms provides us with a convenient, robust, programmable controller which whilst limited compared to some of its industrial counterparts is adequate for this initial proof of concept. The use of LEGO components does bring

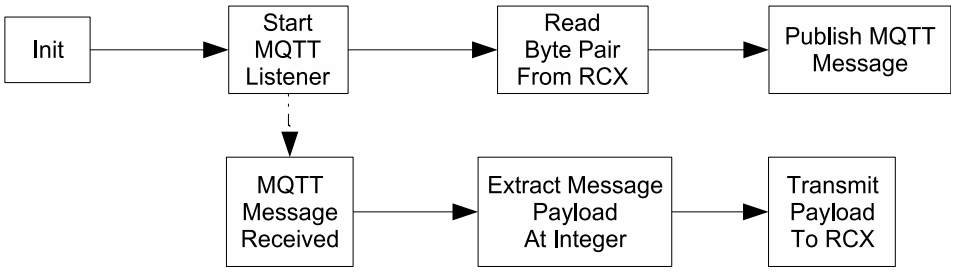


Figure 1: Software flow within the RCX PC Bridge Agent.

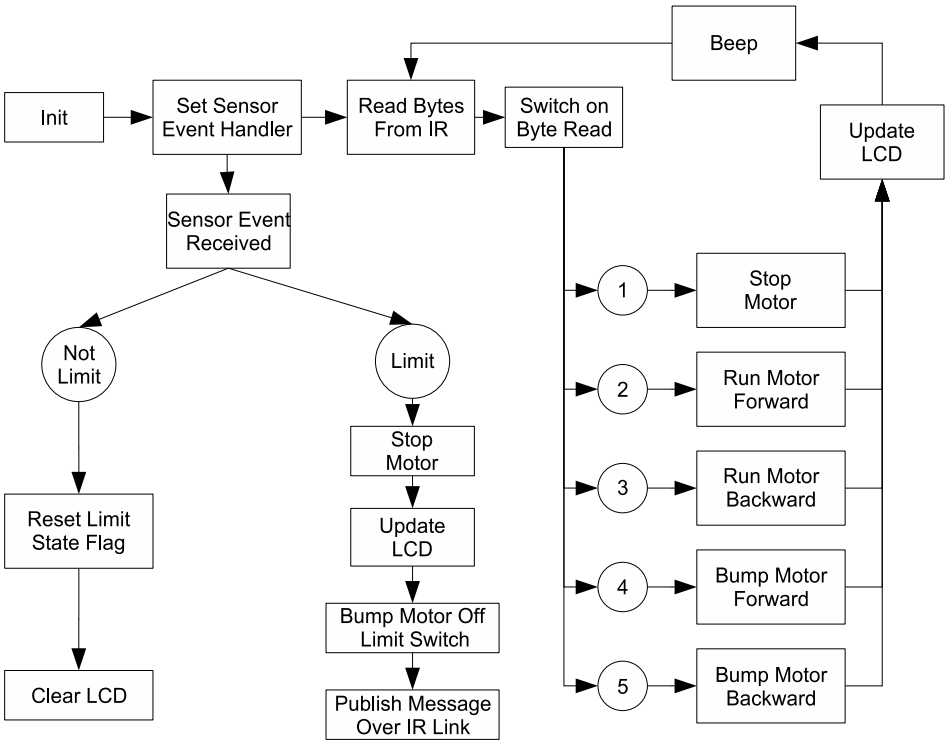


Figure 2: Procedure Flow within the RCX Code.

some limits in terms of repeatabilty and tolerance of construction. For longer term use high spec industrial components and indexed motion controllers would produce a system with a much higher level of reproducability. LEGO also has a limited maximum load capacity (it is after all designed as a toy).

The use of MQTT as a remote control protocol brings some benefits and some unexpected features. By using existing libraries the software development can be extremely rapid, which when producing a demonstrator is important. MQTT also provides a solution to some of the traditional network problems, for example dealing with broken or unreliable connections, and making sure that the message always gets through. However, with remote control it is necessary to consider if messages should always get through, if the end effector is unable to recieve the control signal straight away, should the system then queue it for later delivery? It was found with this work that the reliability of the IR link between the PC and the RCX was somewhat variable, sometimes it was necessary to manually move the RCX to regain signal. When the link was regained, MQTT then ensured that waiting messages were delivered, often resulting in unpredictable behaviour. In the case of the LEGO model this was unexpected, but not dangerous. However, with another system the unpredictable behaviour from delayed messages could cause a dangerous situation, hence control hardware and software should be designed to avoid this, in particular using the non persistent messaging mode of MQTT.

4 Extensions and Future work

4.1 Sample Handling

As part of automating an apparatus in a laboratory moving physical samples onto and off the apparatus needs addressing. Some existing apparatus are fitted with auto samplers and sample batching handlers which allow the operator to preload a number of samples, usually into a grid or carousel that then sequentially presents the samples to the measurement position. By combining sample selection hardware with a scheduling and notification system the remote user could arrange a time on the apparatus during which their sample would be mounted, and they would be given control of the apparatus.

4.2 Inclusion of Remote Monitoring

Work on remote environment and apparatus monitoring has previously reported that MQTT can be used to monitor sensors in realtime(1; 3; 9). When implementing automation on a real-laboratory scenario, it is important to provide the operator with as much information about the apparatus, and personnel in the vicinity of the apparatus as possible, both for reasons of safety and result reliabilty. By using common technologies adding this data becomes a far easier task.

4.3 Authorisation over MQTT

MQTT deliberately has a very minimalist approach to security, enabling appropriate security to be layered on top of it as required for any given application. In this demonstrator, security was purpously overlooked as the equipment is relativly safe, and data traffic limited to the campus network. Before using this type of system to allow control over a hostile network (such as the internet) an appropriate security scheme would need to be implemented. This could include; Encrypting the message payload (which could use PKI certificates to additionally provide signing for authentication and non-repudiation), Challenge/response security (by sending the challenge/response flows as MQTT messages over pub/sub), or TCP/IP traffic protection (such as standard VPN or SSH tunnels).

References

[1] J. M. Robinson, J. G. Frey, A. J. Stanford-Clark, A. D. Reynolds, B. V. Bedi, Sensor networks and grid middleware for laboratory monitoring, in: First International Conference on e-Science and Grid Computing (e-Science'05), pp. 562 – 569, pages.
URL http://doi.ieeecomputersociety.org/10.1109/E-SCIENCE.2005.73

[2] A. Stanford-Clark, Integrating monitoring and telemetry devices as part of enterprise information resources, Tech. rep., IBM (2002).
URL http://www-306.ibm.com/software/integration/mqfamily/integrator/telemetry/pdfs/telemetry_integration_ws.pdf

[3] J. M. Robinson, J. G. Frey, A. J. Stanford-Clark, A. D. Reynolds, B. V. Bedi, Chemistry by mobile phone (or how to justify more time at the bar), in: Proceedings of the UK e-Science All Hands Meeting 2005, , pages.
URL http://www.allhands.org.uk/2005/proceedings/papers/481.pdf

[4] IBM, WebSphere MQ Telemetry Transport Specification.
URL http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp?topic=/com.ibm.etools.mft.doc/ac10840_.htm

[5] Lejos : Java for the rcx, accessed 24 April 2006.
URL http://lejos.sourceforge.net/

[6] Cpia webcam driver for linux, accessed 24 April 2006.
URL http://webcam.sourceforge.net/

[7] A. Nirendra, Exploring procfs, Linux Gazette 115.
URL http://linuxgazette.net/115/nirendra.html

[8] jtravis@p00p.org, Official camserv home page, accessed 28 April 2006.
URL http://cserv.sourceforge.net

[9] Floodnet: Pervasive computing in the environmment, accessed 24 April 2006.
URL http://envisense.org/floodnet/floodnet.htm

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Dynamic Operating Policies for Commercial Hosting Environments

J. Slegers, C. Smith, I. Mitrani, A. van Moorsel and N. Thomas*

July 17, 2006

Abstract

This paper reports on two strands of work that are being undertaken as part of the EPSRC funded DOPCHE project. The paper focuses on open software architectures for dynamic operating policies and a performance model used to find optimal operating policies.

1 Introduction

Amongst the areas of research vital to the development of infrastructure support for eScience is the provision of systems with predictable, differentiated levels of performance and dependability [13]. It is essential that systems be able to provide sustainable levels of service whilst adapting their operating policies to dynamic workloads, resource pools and system configurations. While policy making, prediction and monitoring, and self-tuning infrastructures are available, they exist to different degrees and to different extents, and there is no established framework in which they are (or can be) currently integrated. In future systems it will be imperative that these are provided as fundamental, integrated components of a combined infrastructure.

The success of an eScience infrastructure is based on a number of fundamental requirements, including the ability to provide dynamic, universally available and trusted services. Underpinning such systems is a set of basic functional requirements:

- to understand, capture and define service requirements,
- to verify that the infrastructure is delivering the desired quality of service,
- to dynamically adjust operating policies if the service requirements are not being met.

There is considerable support for the introduction of service-level agreements in Grid computing. These are seen as a mechanism by which work units from a variety of different customers can be arranged and ultimately coordinated though infrastructure-level operating policies. The full exploitation of the infrastructure by a number of different user groups will indeed require several concurrent operating policies running across different virtual organisations and over different geographical sites. It will only be in meeting these predefined policies (and therefore the service-level agreements) that the notion of a trusted ubiquitous system will be established [2, 11]. There are several research areas which support the development and delivery of universally available and trusted services. These include scheduling and reservation, and mechanisms for managing wide-area applications such as caching and data proximity. However, fundamental to these services is the ability to provide a benchmark comparison of one implementation against another, and in so doing be able to recognise and realise the underlying core performance requirements and delivery.

There is a distinct need for research in the areas of:

- systems and application modelling that allow us to predict reliably the behaviour of future eScience infrastructures;
- performance verification that provides evidence-based validation on the delivery of these services;

*School of Computing Science, University of Newcastle, *nigel.thomas@ncl.ac.uk*

- self tuning and adaptive systems that aim to reduce the cost and complexity of managing the infrastructure in the light of changing user needs and changing infrastructure support.

There is a fundamental need for a coordinated investigation into the relationship between all three work areas in delivering trusted ubiquitous systems, particularly in the light of emerging component technologies.

In this paper we present some initial work from the "Fundamental Computer Science for eScience" funded DOPCHE project. Work at Newcastle is focused in two work packages, one on dynamic software architectures and one on performance based policy selection.

2 Open Software Architecture for Dynamic Operating Policies

The evolution of the distributed systems paradigm is yielding an increased requirement to facilitate interactions with heterogeneous stateful resources through a uniform interface. Provision of a uniform interface assures interoperability and loose coupling by providing a consistent set of interaction methods used to access and manipulate the state of all constituent resources. The uniform interface therefore seeks to homogenise interaction with heterogeneous resources.

Figure 1 shows an example interaction with the state of a resource through a uniform interface, requesting the value, V_1, of some property P_1. We denote the request of an interaction as Rq, and the response as Rp. These requests and responses are specialized by including the type in the form Rq_{type} and Rp_{type} respectively. The interaction methods of the uniform interface enable the state of the $Server_1$ resource to be managed using methods consistent with any other resources implementing this uniform interface.

Enabling interaction with stateful resources through a uniform interface provides opportunities in many deployment environments including self-managing [17] and grid [6] systems. The intrinsically heterogeneous resources in these application domains necessitate consistent interaction methods to allow both the flexible integration of, and intercommunication between participant resources in a self-managing system, and the coordinated resource sharing [6] required in a grid system.

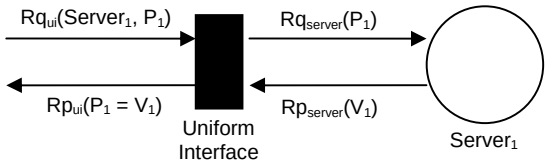


Figure 1: Example uniform interface interaction

The problem we address here is that of providing an efficient uniform interface to enable interaction with and thus management of stateful resources. The efficiency of a uniform interface shall be judged on the syntax required to convey some semantic notion. Thus, an efficient uniform interface for stateful resource interaction would be semantically complete and require minimal syntax. The term resource is used to describe an identifiable object, and we add to this the notion of a resource being stateful. That is, at some time t, the resource r holds some state $S_{t,r}$ characterised by some number of related properties. We define P_r as the set of constituent properties of resource, r, and V as the set of all possible values for these properties. Then, omitting the subscript r for readability, we can define the state S_t of a resource formally as:

$$S_t = \{(p, v) | p \in P \wedge v \in V\}$$

The state S_t represents the evolution of the resource as a result of both time, T, and direct interactions, $i \in I$. The evolution implies the transition from some state S_t to state S_{t+1}.

Figure 2 shows the transition of a given resource from S_t to S_{t+1} as a result of interaction i_1, and from S_{t+1} to S_{t+2} as a result of time. The semantic consequence of the transitions between states is defined by the underlying semantics of the stateful resource. For example, the transition from state S_t to S_{t+1}

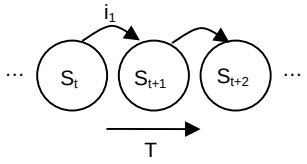


Figure 2: Transitions of a stateful resource

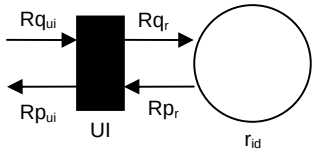


Figure 3: Uniform interface interaction

may amend the value of some property, $p \in P$ (a change to some entity, e, shall be denoted e'). The transition from S_{t+1} to S_{t+2}, conversely, may change the properties composing the resource. The uniform interface should only be concerned with the communication of, and resulting state from the interaction semantics, not their semantic consequence on the represented resource. To represent interactions with a stateful resource, the uniform interface at its most fundamental should supply a set of atomic methods semantically analogous to:

$$M_{ui} = \{GET, SET\}$$

These methods enable complete control over the state of some resource, r; facilitating retrieval (GET) and modification (SET). Interaction with the resource necessitates the assignment of a unique identifier, r_{id}, such that interactions may be directed at resources using this identifier. More specific functionality can be obtained by providing specialisations of these atomic methods, or conveying increased semantic content to these methods. This yields a trade-off between the cardinality of the interaction method set, M_{ui}, and the syntax required by some $m \in M_{ui}$. The most efficient solution to the general problem must resolve this trade-off in an optimal manner.

The fundamental issue to be addressed in a solution is the mapping of the uniform interface interaction to the specific resource interaction, which we define formally as:

$$Rq_{res} = UI(Rq_{ui}) \quad (1)$$

$$Rp_{ui} = UI(Rp_{res}) \quad (2)$$

The conciseness with which one can define the mapping, UI, is dictated by the methods chosen to represent Rq_{ui} and Rp_{ui}. This mapping must be executed by an engine of some sort, and the greater the semantics the engine is able to derive from some given interaction syntax, the more efficient the engine, and consequently the uniform interface. Increased efficiency of a uniform interface has the subsidiary effects of reducing the burden on both the communication protocol used to transport Rq_{ui} and Rp_{ui} between the source and destination resource, and the footprint of the UI function at the interaction endpoints.

2.1 Solution in REST Style

The architecture proposed applies the Representational State Transfer (REST) architectural style, put forward by Fielding [7]. REST declares a set of constraints which focus on achieving desirable properties, such as scalability and reliability within a distributed system. Hypertext Transfer Protocol (HTTP) [7], the archetypal application-level protocol for distributed systems, can be used to enforce the constraints of the REST architecture, facilitating stateless, self-descriptive client-server interactions. HTTP, therefore, acts as the basis for the uniform interface of the proposed solution.

The concept of a resource is central to REST, and is formally defined as a 'temporally varying membership function $M_r(t)$ which for time t maps to a set of entities, or values, which are equivalent' [7]. A resource is conceptual and defined by its mapping to representation at some time t. This definition pertains well to our definition of a stateful resource given above; at time t, $M_r(t)$ will map the resource r to

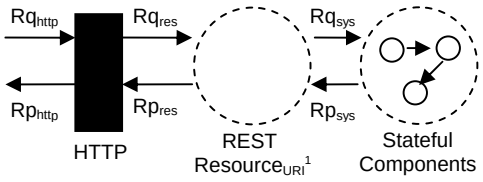


Figure 4: Proposed interface structure

some representation and resource r can be said hold some state S_t.

The notion of a conceptual resource allows us to encapsulate the varying representations (states) of a resource into one conceptual entity, drawing many parallels with the WS-Resource construct in WSRF. For the purpose of entity identification, and indeed for the communication of much of the interaction semantics, we use a Uniform Resource Identifier (URI) [18]. At time t, $M_r(t) = S_t$, and resolve$(URI_1) = S_t$, accordingly the current state representation of the resource can be accessed by resolving the identifier, URI_1. The state representation is composed of all resource properties, P exposed by a resource at some URI, and these properties may be derived from any number of stateful components behind the uniform interface, which once again is comparable with WS-Resource functionality. We define the methods of a uniform interface using HTTP as follows:

$$M_{http} = \{GET, POST, PUT, DELETE\}$$

The interaction methods offered by this uniform interface are defined by HTTP. These methods enable access to (GET), and manipulation of ($POST$, PUT, and $DELETE$) any resource residing at some URI. These methods are communicated using standard HTTP requests, and thus communication is syntactically uncomplicated. The manipulation methods could conceivably be combined into a single $POST$ method, as creation (PUT) and deletion ($DELETE$) of state could be seen as forms of amendment. This would concatenate the method set, but would require additional semantics to be conveyed implicitly in the request body, rather than explicitly in the method line of the request. For syntactic and semantic simplicity we retain all standard HTTP methods shown above.

Introspection on, and amendment to the resource state can be performed through the utilisation of basic HTTP methods and resource URI. The resulting state of a transition can be defined formally as: $S_{t+1} = (S_t \cup (p_i, v_i))\backslash(p_i, v_i)$, where $p_i \in P_r \wedge p_i \in P_r$. State transition is where this uniform interface greatly simplifies any previous solutions.

Any of the interaction methods, $m \in M_{http}$, may include metadata in a response, which is used for the communication of additional information regarding the state. For example, included in the metadata of a GET would be the URIs for access to, and manipulation of component properties of the state, using the standard interaction methods. Supplementary metadata may include XML Schema [19] to define the structure of interaction data. Properties are therefore treated as conceptual resources, and we can partition state into components using URIs, simplifying interactions markedly. For instance (Figure 5), if some resource has properties $\{p_1, p_2\}$, GET to URI_1 will return $S_t = \{(p_1, v_1), (p_2, v_2)\} \cup \{URI_2, URI_3\}$. The URIs are not parts of the state of resource r, they are simply metadata to communicate how component properties may be changed. Interactions can then be directed at these URIs to access or manipulate the associated property.

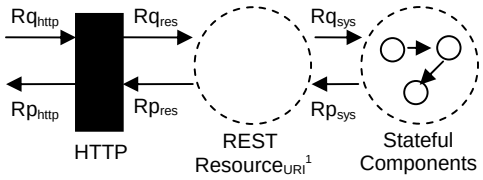


Figure 5: Example interaction in REST

If multiple instances of a resource need to be managed, then we simply incorporate this semantic notion into some URI. We could, for example, define a resource which deals with the management of resources; by performing a GET we can view all managed resources, by performing a PUT we would in-

troduce a new managed resource instance and be returned the URI of the newly created resource for future interactions. The state representation of the management resource would be amended accordingly. This resource notion enables the modelling of any identifiable object whether conceptual or concrete, giving us unbounded modelling flexibility.

$$Rqres = HTTP(Rqhttp) \tag{3}$$

$$Rphttp = HTTP(Rpres) \tag{4}$$

The mapping function, HTTP simply inspects the destination URI and HTTP method of the incoming request. If such a resource exists, and this method of interaction is supported, the HTTP request is forwarded to the resource to execute the internal semantics and respond. If URI does not exist or the method of interaction is not supported, the conventional HTTP response codes are used. We have thus improved the conciseness of the mapping function markedly. All necessary information for interaction is packaged into the HTTP request and URI, no additional contextual information is held on the server relating to a given sequence of interactions. Any context concerning a given (sequence of) interaction(s) is incorporated into the URI and held on the client-side. This adheres to the REST constraint of stateless interactions. The abstraction provided by the concept of a resource in REST enables functionality analogous to the WS-Resource in WSRF. The low level composition of state is again shifted outside of the scope of the uniform interface, relieving much of the semantic and syntactic burden and enabling increasingly flexible notions to be represented as resources. By exploiting the semantics of HTTP and URI, the beneficial effects of abstraction on the uniform interface have been further developed. There have been other attempts to utilise HTTP for the expression of resource interaction semantics [16, 3], but none have fully exploited the benefits of HTTP and URI semantics. Concentration has been on manipulation of HTTP requests and utilisation of URIs to communicate the semantics of some underlying management standard, for instance SNMP [16]. Such solutions are ineffective as they must undergo two stages of processing, from HTTP and URI to the underlying management standard and from this standard to the resource-specific interaction. The resulting architectures therefore demonstrate many of the restrictions of the underlying management standards, albeit while utilising a different communication medium. The REST architecture we propose, through exploitation of HTTP method and URI semantics, enables semantically complete interaction with stateful resources. The abstraction provided by the resource concept enables the reduction in cardinality of the interaction method set, and the utilisation of HTTP methods to convey interaction semantics enables syntactic brevity in interaction. Instead of placing a large amount of syntactic burden on the transport protocol, and mapping function to convey the semantics of the interaction, we are deriving all required semantics from the URI, HTTP method, and any content associated with the request (in some standard data format).

3 Optimal resource provisioning of application hosting in a changing environment

Recent developments in distributed and grid computing have facilitated the hosting of service provisioning systems on clusters of computers. Users do not have to specify the server on which their requests (or 'jobs') are going to be executed. Rather, jobs of different types are submitted to a central dispatcher, which sends them for execution to one of the available servers. Typically, the job streams are bursty, i.e. they consist of alternating 'on' and 'off' periods during which demands of the corresponding type do and do not arrive.

In such an environment it is important, both to the users and the service provider, to have an efficient policy for allocating servers to the various job types. One may consider a static policy whereby a fixed number of servers is assigned to each job type, regardless of queue sizes or phases of arrival streams. Alternatively, the policy may be dynamic and allow servers to be reallocated from one type of service to

another when the former becomes under-subscribed and the latter over-subscribed. However, each server reconfiguration takes time, and during it the server is not available to run jobs; hence, a dynamic policy must involve a careful calculation of possible gains and losses.

The purpose of this work is to (i) provide a computational procedure for determining the optimal static allocation policy and (ii) suggest acceptable heuristic policies for dynamic server reconfiguration. In order to achieve (i), an exact solution is obtained for an isolated queue with n parallel servers and an on/off source. The dynamic heuristics are evaluated by simulation.

The problem described here has not, to our knowledge, been addressed before. Much of the server allocation literature deals with polling systems, where a single server attends to several queues [4, 5, 8, 9, 10]. Even in those cases it has been found that the presence of non-zero switching times makes the optimal policy very difficult to characterise and necessitates the consideration of heuristics. The only general result for multiple servers concerns the case of Poisson arrivals and no switching times or costs: then the $c\mu$-rule is optimal, i.e. the best policy is to give absolute preemptive priority to the job type for which the product of holding cost and service rate is largest (Buyukkoc et al [1]).

A model similar to ours was examined by Palmer and Mitrani [12]; however, there all arrival processes were assumed to be Poisson; also, the static allocation was not done in an optimal manner. The novelty of the present study lies in the inclusion of on/off sources, the computation of the optimal static policy and the introduction of new dynamic heuristics.

3.1 The model

The system contains N servers, each of which may be allocated to the service of any of M job types. There is a separate unbounded queue for each type. Jobs of type i arrive according to an independent interrupted Poisson process with on-periods distributed exponentially with mean $1/\xi_i$, off-periods distributed exponentially with mean $1/\eta_i$ and arrival rate during on-periods λ_i $(i = 1, 2, ..., M)$. The required service times for type i are distributed exponentially with mean $1/\mu_i$. This model is illustrated in Figure 6.

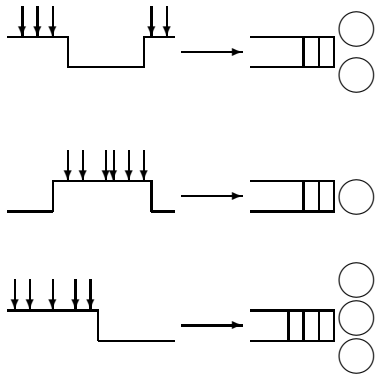


Figure 6: Heterogeneous clusters with on/off sources

Any of queue i's servers may at any time be switched to queue j; the reconfiguration period, during which the server cannot serve jobs, is distributed exponentially with mean $1/\zeta_{i,j}$. If a service is preempted by the switch, it is eventually resumed from the point of interruption.

The cost of keeping a type i job in the system is c_i per unit time $(i = 1, 2, ..., M)$. These 'holding' costs reflect the relative importance, or willingness to wait, of the M job types. The system performance is measured by the total average cost, C, incurred per unit time:

$$C = \sum_{i=1}^{N} c_i L_i \ , \tag{5}$$

where L_i is the steady-state average number of type i jobs present. Those quantities depend, of course, on the server allocation policy.

In principle, it is possible to compute the optimal dynamic switching policy by treating the model as a Markov decision process and solving the corresponding dynamic programming equations. However, such a computation is tractable only for very small systems. What makes the problem difficult is the size of the state space one has to deal with. The system state at any point in time is described by a quadruple, $S = (\mathbf{j}, \mathbf{n}, \mathbf{u}, \mathbf{m})$, where $\mathbf{j}$ is a vector whose ith

element, j_i, is the number of jobs in queue i (including the jobs in service); $\mathbf{n}$ is a vector whose ith element, n_i, is the number of servers currently assigned to queue i; $\mathbf{u}$ is a vector whose ith element, u_i, is 0 if the ith arrival process is in an off-period, 1 if it is on; $\mathbf{m}$ is a matrix whose element $m_{i,k}$ is the number of servers currently being switched from queue i to queue k. The possible actions that the policy may take in each state are to do nothing or to initiate a switch of a server from queue i to queue k.

A numerical procedure to determine the optimal policy would involve truncating the queue sizes to some reasonable level, discretizing the time parameter through uniformization and then applying either policy improvement or value iterations (e.g., see [14, 15]). It is readily appreciated that the computational complexity of that task grows very quickly with the number of queues, M, the number of servers, N, and the truncation level. For that reason, we have concentrated on determining the optimal static allocation policy (which does not involve switching) and comparing its performance with that of some dynamic heuristics.

3.2 Initial results

Figure 7 shows the total (i.e. for both job types combined) average (i.e. over time) cost, over a period of $T = 10000$, of a system with 2 job types and 20 servers. Both the job types are on for half the time but one of them has a cycle lasting (in mean) 50 and the other 100. The holding cost for the first job type is 1, for the second it is 2. In all other aspects they are identical and the rate with which jobs are completed per server is 1. The x-axis represents increased load, with the rate of job arrival increasing from 15 to 19.5. This means the system is only stable on average. The switching time and cost is set to be very small in this case. Heuristic 1 is making switching decisions based on a fluid cost approximation until the next time the queue empties. The assumption here being that job types that are on, remain on and vice versa. Heuristic 2 is the same as above, but using the average load to phase the on-off periods out of the model.

Because of time constraints, these are averages over a small number of simulations. This explains some

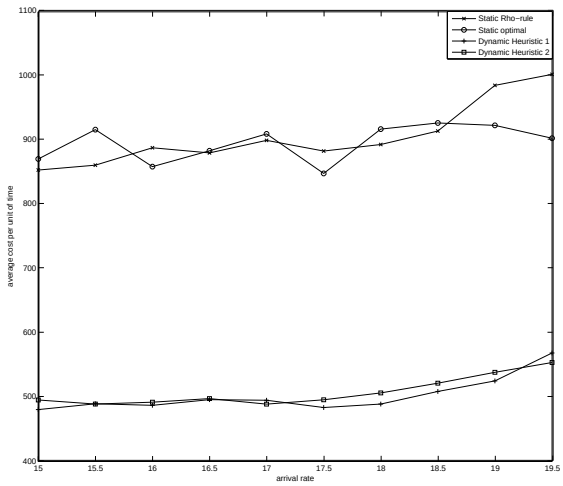


Figure 7: Total average cost, over a period of T=10000, of a system with 2 job types and 20 servers

big variations, especially in the static cases. Note that it is probably not a coincidence that the dynamic simulations seem to fluctuate much less, their very nature makes them more capable of dealing with more extreme events. Note that the dynamic policies are a big improvement over the static ones studied previously [12].

4 Conclusions

In this paper we have presented some preliminary results concerning issues relating to dynamic deployment within commercial hosting environments. These results show the suitability of using the REST architectural style for developing an open software architecture for dynamic operating policies. In addition we have shown some initial results from modelling experiments using stochastic modelling techniques to derive improved policy decisions. These results show that where the requests for a service fluctuate, taking this additional information into account when assigning servers can have a significant benefit.

A substantial amount of further work remains to be done within the DOPCHE project in developing optimal and heuristic policies and in developing the

architectural support to implement these policies in practise.

References

[1] C. Buyukkoc, P. Varaiya and J. Walrand, "The $c\mu$-rule revisited", *Advances in Applied Probability*, 17, pp 237-238, 1985.

[2] K. Czajkowski, I. Foster, C. Kesselman, V. Sander, S Tuecke, SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed Systems, 8th International Workshop on Job Scheduling Strategies for Parallel Processing, LNCS 2537: 153-183, Springer- Verlag, 2002.

[3] L. Deri, Surfing Network Resources across the Web, In Proceedings Of 2nd IEEE Workshop on Systems Management, 1996, pp. 158-167

[4] I. Duenyas and M.P. Van Oyen, "Heuristic Scheduling of Parallel Heterogeneous Queues with Set-Ups", *Technical Report 92-60*, Department of Industrial and Operations Engineering, University of Michigan, 1992.

[5] I. Duenyas and M.P. Van Oyen, "Stochastic Scheduling of Parallel Queues with Set-Up Costs", *Queueing Systems Theory and Applications*, 19, pp 421-444, 1995.

[6] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of High Performance Computing Applications, 15(3), 2001, pp. 200-222

[7] R.T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, PhD thesis, Department Of Information and Computer Science, University Of California, Irvine, 2000.

[8] G. Koole, "Assigning a Single Server to Inhomogeneous Queues with Switching Costs", *Theoretical Computer Science*, 182, pp 203-216, 1997.

[9] G. Koole, "Structural Results for the Control of Queueing Systems using Event-Based Dynamic Programming", *Queueing Systems Theory and Applications*, 30, pp 323-339, 1998.

[10] Z. Liu, P. Nain, and D. Towsley, "On Optimal Polling Policies", *Queueing Systems Theory and Applications*, 11, pp 59-83, 1992.

[11] H. Ludwig, A. Keller, A. Dan, R. King, A Service Level Agreement Language for Dynamic Electronic Services, 4th IEEE International Workshop on Advanced Issues of E-Commerce and Wen-based Information Systems, IEEE Computer Society Press, 2002.

[12] J. Palmer and I. Mitrani, "Optimal Server Allocation in Reconfigurable Clusters with Multiple Job Types", Journal of Parallel and Distributed Computing, 65/10, pp 1204-1211, 2005.

[13] J. Schopf, M. Baker, F. Berman, J. Dongarra, I. Foster, B. Gropp, T. Hey, "Grid Performance Workshop Report", whitepaper from the International Grid Performance Workshop, May 12-13, UCL, London, 2004.

[14] E. de Souza e Silva and H.R. Gail, "The Uniformization Method in Performability Analysis", in *Performability Modelling* (eds B.R. Haverkort, R. Marie, G. Rubino and K. Trivedi), Wiley, 2001.

[15] H.C. Tijms, Stochastic Models, Wiley, New York, 1994.

[16] C. Tsai, R. Chang, SNMP through WWW, International Journal Of Network Management, 8(1), 1998, pp.104-119

[17] A. van Moorsel, Grid, Management and Self-Management, The Computer Journal, 48(3), 2005, pp.325-332

[18] W3C, Uniform Resource Identifier (URI): Generic Syntax, ed. T. Berners-Lee, R.T. Fielding, L. Masinter, 1998

[19] W3C, XML Schema, ed. H.Thompson, D. Beech et al, 2004

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

A framework for Grid-based failure detection in an automated laboratory robot system

C. Foulston and A. Clare

Department of Computer Science, University of Wales Aberystwyth,
Penglais, Aberystwyth SY23 3DB
afc@aber.ac.uk

Abstract

We have designed and produced a framework for Grid-based failure detection to monitor and report on our automated laboratory equipment and the Robot Scientist. Its features are distributed agent based monitoring, selective reporting and report dispatch brokering. Monitoring and reporting agents can be distributed due to the loose coupling of Web Services, enabling a large enviroment of parameters to be monitored. Reporting is via different mediums such as e-mail, instant PC alerts and text messaging, though any type of agent can sign up for new reports, making the system expandable to a variety of needs. Dispatch brokering allows agents and humans to sign up for the latest reports for further analysis.

1 Introduction

The "Robot Scientist" is a state-of-the-art e-Science system for automating the scientific process [3]. It comprises automated and integrated laboratory equipment together with intelligent software for creating hypotheses, designing the high throughput experiments and analysing the results. The system is currently used to conduct yeast mutant growth experiments in order to investigate gene function in metabolic pathways. The intelligent software provides closed loop learning, whereby the results of the previous experiment are fed back into the system in order to refine hypotheses and choose and conduct the next round of experiments automatically.

The system must function unaided for long periods of time. The average yeast growth period measured will be 5 days, and experimental batches are overlapped, so that operation is continuous. The equipment is located in a laboratory in a different room to that in which the Robot Scientist project researchers work. The yeast is grown up in a pregrowth phase lasting approximately 24 hours, which is followed by a growth phase lasting a couple of days. Important events may happen at any time of day or night, and our lab technician will not always be around to watch.

We need an remotely accessible automated failure detection system for the Robot Scientist. This should:

- monitor equipment
- log information, errors and warnings
- take intelligent decisions and act upon them
- notify users of problems by a variety of means
- provide users with information (current and historical) and suggest possible actions to be taken
- keep records of previous problems, solutions and actions

In order to make intelligent decisions about whether a potential failure has been detected, and what action to take, we need to employ a variety of reasoning methods in the system, analysing a wide range of data. The detection processes are discussed in Section 3.

The failure detection system has to be available to monitor remotely from a variety of OS platforms, and it has to be secure and reliable. Providing Web Services interfaces to its functionality is a good way to achieve this, and we chose to use the Globus toolkit [2] to provide some of the basic infrastructure. The interfaces and general software architecture is discussed in Section 4.

2 Background

2.1 Robot Scientist

The Robot Scientist uses a large collection of laboratory automation equipment to conduct the experiments that are designed by the artifical intelligence software. The equipment is capable of growing yeast knockout mutant strains under a variety of experimental conditions. Over 1000 experiments can be performed each day, and the main experimental outputs are optical density readings that are used to plot the growth curves of the yeast strains. A large amount of experimental metadata

is also available from each component of the lab automation equipment.

The robot components are controlled and coordinated by software. The whole system is continually supplied with descriptions of experiments that have been designed to test hypotheses created by AI software. The system should run continuously, 24 hours a day, 7 days a week.

Any failure of this system needs to be detected as early as possible, so that action can be taken, and the equipment can be used to its maximum potential. A single fault can cause the entire system to be unusable, and all experiments currently running (which may have been running for days) may need to be abandoned. Faults can range from an obvious show-stopping breakdown to a slight variation of some condition, which could still have a disastrous effect on the results of experiments.

2.2 Existing e-Science labs and failure detection systems

Very little laboratory-based and experimental equipment is actually Grid-enabled yet. The Grid is still an evolving concept and toolkits such as Globus have not yet matured enough for most industrial use.

Ko et al [4] describe a laboratory with a web-based interface, designed to be accessed remotely by students to run coupled tank control experiments. The system provides feedback to the students, by video, audio and by data, such as plots of response curves. The students can access the remote laboratory 24 hours a day. The potential of remote biology labs used for education is explored further by Che [1].

NEESgrid is a large grid-based system for earthquake simulation and experimentation. Laboratories are linked via grid infrastructure to compute resources, data resources and equipment (`http://it.nees.org/`). Various instruments and sensors can be monitored and viewed remotely, and NEESgrid provides teleobservation and telepresence via video streams, data streams and still images.

The DAME project is a major e-Science pilot project providing a distributed decision support system for aircraft maintenance. They proposed a Grid based framework for fault diagnosis and a implementation for gas turbine engines [5]. Their work is perhaps the most similar to ours, though for a very different application, and used Globus Toolkit version 3 to provide web service interfaces. They note the benefits of using a loosely coupled service-oriented architecture to implement a variety of fault diagnostic services. A engine simulation service, case based reasoning services and event sequence/history monitoring services are provided.

Otherwise, fault detection in Grid-based systems has so far been mostly targeted towards analysis of network faults, and detection of problems in compute clusters. Tools for monitoring and fault detection exist, but are currently mostly restricted to monitoring network traffic and general performance of Grid services (response time, availability, etc). Globus has a Monitoring and Discovery System (MDS) component that allows querying and subscription to published data. It is intended to be an interface to other monitoring systems, and to allow a standard interface to the data. Other general grid service monitoring tools include Gridmon[1], Network Weather Service[2], Netlogger[3], Inca[4] and Active Harmony[5]. An annual Grid Performance Workshop[6] reflects current research in these systems.

Our laboratory automation system is complex and contains many potential sources of failure, including human error, hardware error and experimental error. In the next section we describe these and the methods of failure detection in more detail.

3 System requirements

3.1 Sources of information

The primary sources of information for use in detecting failure will be:

Equipment metadata logs: Each piece of equipment logs information such as event timings and internal settings and variation.

Experimental data/observations: The optical density readings of the yeast can show that something is wrong. For example, an oscillating pattern of readings for a microtitre plate lead us to discover that the two incubators between which the plate was being cycled were set to agitate at different speeds.

Experimental expectations: The Robot Scientist work is perhaps unique in having the whole process automated, so we can make use of the fact that the expected result of every experiment is automatically available for comparison to the actual result. Of course, an unexpected result may also be due to the discovery of new scientific knowledge.

[1] Gridmon: `http://www.gridmon.dl.ac.uk/`
[2] NWS: `http://nws.cs.ucsb.edu/`
[3] Netlogger: `http://www-didc.lbl.gov/NetLogger/`
[4] Inca: `http://inca.sdsc.edu/`
[5] Active Harmony: `http://www.dyninst.org/harmony/`
[6] Grid Performance Workshop: `http://www-unix.mcs.anl.gov/~schopf/GPW2005/`

Temperature/humidity sensors: These are scattered throughout the system, monitoring the ambient conditions, rather than instrument-specific conditions.

Webcams: In the future, image recognition should be able to distinguish some fault states, and provide more information on the causes of robotic arms becoming blocked, or tips being dropped. Experiment imaging, such as closeups of plates or cells could also provide valuable information.

3.2 Types of failure

Hardware error: The equipment consists of many integrated automated lab machines. These include incubators, liquid handlers, a washing station, a centrifuge, microplate readers and several robot arms and shuttles for moving plates around the system. Each of these components has a software interface which can report errors that the machine can detect (such as being wrongly initialised, overheating, being full, or dropping what it was carrying). We have already experienced power cuts, freezer motor seizure, dropped tips, misaligned robot arms, and a host of other hardware issues. Human error can also contribute to hardware reports. If the technician did not provide the system with the enough plates for the experiment then at some point there will be none left on the stack to take (and similarly for emptying of waste). While human errors are mostly avoidable, they still occur. Hardware errors due to imperfections in plastic tip mouldings, cable stretch over time, loss of vaccuum suction, etc, are much more difficult to avoid.

Experimental error: Some errors may be less obvious, and may only be noticeable when the experimental results are interpreted in context, or compared to some model of what was expected. For example, if the liquid hander is not aspirating the quantity of liquid that it claims to have aspirated, this may show up in altered plate readings. Similarly, if the plate readings were not as expected this could also point to altered growth temperature, contamination or misplaced strains. If wells at one side of a plate grew less well than wells at the other side it might be suspected that conditions across the plate were uneven, for example air flow, oxygen availability, evaporation or temperature.

Software error: Software errors are much more difficult to detect. There is a huge field of research devoted to the design of error-free software, incuding methods for debugging, refactoring, formal specification, and code development practices. With existing, pre-installed software our aim is simply to detect if a fault has occurred that is likely to have a software origin so that it can be fixed. However, these can be very complex and manifest in unusual ways.

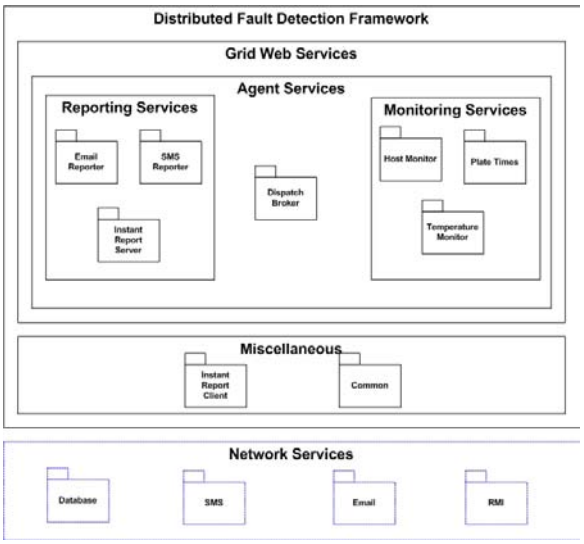


Figure 1: Framework overview, with examples of the Monitoring Agents

3.3 Detection Methods

A variety of different methods will be necessary to detect the different types of faults. These will include case based systems, model based systems, use of historical data, use of experimental expectations and hypotheses, use of event data and precise timings, knowledge of stock control of nutrients, yeast strains and growth media, use of hardware logs and error codes, and many other analyses.

4 Implementation

4.1 Framework

The framework is abstract enough to be applicable to most lab automation systems. Figure 1 shows the components of the framework.

4.2 Reporting Agents

The current reporting agents that bind to the dispatch broker are a text message reporter, an email reporter and an instant desktop reporting tool. Reporting to a user's machine is the quickest way but users may not always be at their desks. Therefore text messaging is an important method for urgently attracting attention. Selective reporting is used here to specify times for this as researchers are not always on duty.

4.3 Abstract Monitoring Agents

Example implementations of the abstract monitor are:

- The incubator monitor, which monitors temperature, humidity, O_2 and CO_2 levels of three incubators. This reports to the dispatch broker when thresholds are not met. The yeast must be grown in a stable, controlled and measurable enviroment. Only with a wide array of different environmental state monitors will it be possible to ensure all experiments were done in a similar enviroment.
- A host monitor which simply monitors the main control PC running the robot. The host monitor's job is to ensure that all control PCs are up and running at all times and to report this as a high alert to the dispatch broker. When PCs fail this will waste experimental time and most likely ruin current experiments.

Monitoring Agents are separated from the business of making decisions about who to report to or how to report by the dispatch broker. This allows a Monitoring Agent to focus just on the detail of monitoring. This also allows Monitoring Agents to be composed of other Monitoring Agents in a hierarchical manner. A more complex Monitor that relies on several aspects of the system can be built using the results of several low level Monitoring Agents, hence avoiding time-consuming repetition of low level tests.

Parameters that need monitoring come from many different systems such as databases, file-based data logs, computational tasks or querying serial or USB interfaces to discover continuous parameters, and these may not be in close quarters, therefore it is essential to have a loose coupling for agents.

4.4 Dispatch Broker

The dispatch broker recives reports from any agent and notifies all outbound agents that a new report has been delivered. This ensures the loose coupling between detecting and reporting.

4.5 Selective Reporting

This allows users to select which reports they wish to receive.

This is to ensure that information passed on is relevant in the context of the user it is sent to. On such a large system as the Robot Scientist, researchers with completely different backgrounds will be interested in different reports. Biologists may not want to concern themselves with computer hardware faults and computer science researchers may not want to concern themselves with warnings about low liquid levels. This uses a relational database where the relations between researchers and monitors are stored.

5 Discussion

Detecting and recording errors and failures is essential for any highly complex system of many different parts. Even fully automated biological experiments are prone to noise, and careful monitoring of conditions and experimental metadata is crucial for correct interpretion of results. We need to be able to detect problems as they occur, rather than days later when valuable experimental time and resources have been wasted.

The failure detection system that has been developed is general enough to be applicable to most laboratory automation environments. It uses Globus/Web Services to provide a loose coupling of components. It is in place on the Robot Scientist with the most immediately essential monitoring agents, and we now need to extend its capabilities with a wide variety of more complex monitoring agents. More intelligent agents will provide diagnosis as well as detection, and intelligent, user-friendly diagnosis of biological problems will be an interesting area of research in its own right.

Acknowledgements

The authors would like to thank Dr Andrew Sparkes for valuable discussions.

References

[1] A. Che. Remote biology labs. In *E-ducation Without Borders*, 2005.

[2] I. Foster. Globus toolkit version 4: Software for service-oriented systems. In *IFIP International Conference on Network and Parallel Computing*, pages 2–13. Springer-Verlag LNCS 3779, 2005.

[3] R. D. King, , K. E. Whelan, F. M. Jones, P. G. K. Reiser, C. H. Bryant, S. Muggleton, D. B. Kell, and S. G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 2004.

[4] C. C. Ko, B. M. Chen, J. Chen, Y. Zhuang, and K. C Tan. Development of a web-based laboratory for control experiments on a coupled tank apparatus. *IEEE Transactions on Education*, 44(1), 2001.

[5] X. Ren, M. Ong, G. Allan, V. Kadirkamanathan, H. A. Thompson, and P. J. Fleming. Service oriented architecture on the Grid for FDI integration. In *Proc 3rd UK e-Science All Hands Meeting (AHM 2004)*, 2004.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Deciding semantic matching of stateless services

Andrey Bovykin[1] and **Evgeny Zolin**[2]

[1] Department of Computer Science, University of Liverpool, UK, andrey@csc.liv.ac.uk

[2] School of Computer Science, University of Manchester, UK, ezolin@cs.man.ac.uk

Abstract

We present a novel approach to describe and reason about stateless information processing Semantic Web Services. It can be seen as an extension of standard descriptions which makes explicit the relationship between inputs and outputs and takes into account OWL ontologies to fix the meaning of the terms used in a service description. This allows us to define a notion of *matching* between services that yields high precision and recall for service location. We explain why matching of these kinds of services is *decidable*, and provide examples of biomedical web services to illustrate the utility of our approach.

1 Introduction

Many of the tools and databases for analysing the data generated from genome sequencing projects like the Human Genome Project are available via Web Service interfaces. They allow biomedical scientists to use the Web as a platform to perform so-called *in silico* experiments. Large numbers of *in silico* experiments are carried out by choosing some of these Web Services, composing them into a workflow, and running them—an approach which shows considerable promise for molecular biology [3] whilst challenging current Web Service approaches. Nowadays, a wide variety of domain ontologies exist which capture the knowledge of biologists (see http://obo.sourceforge.net/).

The majority of these services are *stateless*, i.e., they provide information, but do not change the state of the world. Hence their descriptions do not need to include pre- and post-conditions. Here we restrict our attention to these kinds of services since they are quite common yet easier to represent, and since it turned out that defining a semantics or specifying automated reasoning algorithms for world-altering services is basically impossible in the presence of any expressive ontology [9].

The question we are interested in here is how to help a biologist to find a service he or she is looking for, i.e., a service that works with inputs and outputs the biologist can provide/accept, and that provides the required functionality. The growing number of publicly available biomedical web services, 3 000 as of February 2006, requires better matching techniques to locate services. Thus, we are concerned with the question of how to describe a service request Q and service advertisements S_i such that the notion of a service S *matching* the request Q can be defined in a "useful" way. By useful, we mean the following: (1, precision) only those services should match the request that indeed provide the requested functionality; (2, recall) all services providing the requested functionality should match the request; (3) service advertisements and requests should be formulated using terms from existing (OWL) ontologies; and (4) such that the matching problem can be decided automatically.

In this paper, we will propose *a framework to describe in formation providing stateless services that takes into account background ontologies and that allows services to be matched automatically with high precision and high recall.*

2 Services as queries

From a syntactic viewpoint, this framework can be seen as an extension of the way services are described in the OWL-S Service Profile (namely, of its part concerning description of inputs and outputs). Semantically, the service matching conditions we formulate yield the service discovery of higher precision and recall.

In our framework, a description of a service contains, in addition to the types of its inputs and outputs, an explicit specification of the *relationships* between them. Analysing numerous examples of services—including those in bioin-

formatics, see Section 3—it was observed that the notion of *conjunctive query* can be adopted for these purposes. Before introducing this "services as queries" approach formally, let us illustrate it with a simple example from [6] (realistic examples from the bio-informatics domain are given in Section 3).

Let S_1 and S_2 be services both having an input of type GeoRegion and an output of type Wine, and suppose that S_1 (resp., S_2) returns the list of wines that are *produced* (resp., *sold*) in the region with which it was called. If the types of inputs and outputs are the only information available to match a request to a service, then no matching algorithm can distinguish between S_1 and S_2, and thus matching cannot be precise—see (1) above.

Next, assume that a user requests a service Q that takes a FrenchGeoRegion as input and returns the list of FrenchWines that are *produced* in this region. Even though the service S_1 returns, in general, wines that may not be FrenchWines, it returns only FrenchWines when called with a FrenchGeoRegion, and thus should be matched to this request. A matching algorithm that does so has a high recall—see (2) above.

More formally, when run with a GeoRegion `g`, the service S_1 returns all those wines `w` *for which there exists some winegrower* `f` *who produces* `w` *and who is located in* `g`. In our framework, this service can be described as follows:

```
INPUT  g:GeoRegion
OUTPUT w:Wine
THERE IS SOME f   [ WineGrower(f),
      LocatedIn(f,g), Produces(f,w) ]
```

The terms `Wine`, `LocatedIn`, etc., are defined in some ontology. In contrast, the service S_2 returns all those wines `w` *for which there exists some shop* `s` *who sells* `w` *and who is located in* `g`, and can thus be described as follows:

```
INPUT  g:GeoRegion
OUTPUT w:Wine
THERE IS SOME s
 [ Shop(s), LocatedIn(s,g), Sells(s,w) ]
```

In what follows we show that matching service descriptions of this kind is reducible to query containment w.r.t. an ontology—a task whose decidability and complexity is relatively well understood.

2.1 Describing services

We assume the reader to be familiar with OWL-DL and its semantics [11]. Throughout this paper, we borrow the term *TBox* for a class-level ontology (i.e., a finite set of OWL-DL axioms) and *ABox* for a factual ontology (i.e., a finite set of OWL-DL facts). The union of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ is called a *knowledge base* and denoted by $\mathcal{KB} = \langle \mathcal{T}, \mathcal{A} \rangle$. We use $\mathcal{KB} \models \Psi$ to denote the fact that $\mathcal{KB}$ implies Ψ, i.e., Ψ holds in every interpretation that satisfies $\mathcal{KB}$.

Definition 1 (Service syntax). A *service description* $S{=}\langle \vec{x}:\vec{X}; \vec{y}:\vec{Y}; \Phi(\vec{x},\vec{y},\vec{z}) \rangle$ consists of

- a list $\vec{x}:\vec{X} = \langle x_1{:}X_1, \ldots, x_m{:}X_m \rangle$ of pairs of variables x_i and classes X_i; this list enumerates *input* variables and their "types";
- a list $\vec{y}:\vec{Y} = \langle y_1{:}Y_1, \ldots, y_n{:}Y_n \rangle$ of pairs of variables y_j and classes Y_j; this list enumerates *output* variables and their "types";
- a *relationship specification* Φ of the form

  $$term_1(\vec{x},\vec{y},\vec{z}) \wedge \ldots \wedge term_k(\vec{x},\vec{y},\vec{z}),$$

  where each $term_i(\vec{x},\vec{y},\vec{z})$ is either an expression of the form $w{:}C$ with C a class or wRw' with R a property, and w, w' variables from $\vec{x},\vec{y},\vec{z}$ or individual names.

Definition 2 (Service semantics). A service s *implements* a service description S over a TBox $\mathcal{T}$ if, for any ABox $\mathcal{A}$ and any tuple of individuals $\vec{a}$ in $\mathcal{T},\mathcal{A}$, if $\mathcal{T},\mathcal{A} \models \vec{a}:\vec{X}$, then

1. s accepts $\vec{a}$ as input and
2. when run with $\vec{a}$ as input, it returns the set of all those tuples of individuals $\vec{b}$ from $\mathcal{A}$ such that $\mathcal{T},\mathcal{A} \models \vec{b}:\vec{Y} \wedge \exists \vec{z}\, \Phi(\vec{a},\vec{b},\vec{z})$.

Intuitively, this means that the service s must accept all tuples $\vec{a}$ that conform the input type $\vec{X}$ declared in S, and return as its output the set of those instances $\vec{b}$ of the output type $\vec{Y}$ that bear the relationship Φ to the input tuple $\vec{a}$.

2.2 Matching services

Matching is the problem of determining whether a given service description S conforms to another service description Q. Matching algorithms can be used for *service discovery* purpose, and we can think of S as being a service advertisement and of Q as being a service requested by a user. Let us first give a formal definition and then provide explanations. Here we use $|\vec{x}|$ to denote the length of a vector $\vec{x}$.

Definition 3. Given two service descriptions:

$$S = \langle\, \vec{x}:\vec{X};\ \ \vec{y}:\vec{Y};\ \ \Phi(\vec{x},\vec{y},\vec{u})\,\rangle,$$
$$Q = \langle\, \vec{z}:\vec{Z};\ \ \vec{w}:\vec{W};\ \ \Psi(\vec{z},\vec{w},\vec{v})\,\rangle,$$

with $|\vec{x}| = m = |\vec{z}|$ and $|\vec{y}| = n = |\vec{w}|$, we say that the service S *matches* the request Q w.r.t. the TBox $\mathcal{T}$ if there exist two permutations

$$\pi\colon \{1,\dots,m\} \to \{1,\dots,m\}$$
$$\rho\colon \{1,\dots,n\} \to \{1,\dots,n\}$$

such that the following two conditions hold:

(i) $\mathcal{T} \models Z_{\pi(i)} \sqsubseteq X_i$, for all $1 \leqslant i \leqslant m$.

Intuitively, this means that we can map the inputs from S to the inputs from Q such that all input data that the user intends to provide will be accepted by S.

(ii) for any ABox $\mathcal{A}$ and any individuals $\vec{a}, \vec{b}$ in the knowledge base $\mathcal{KB} = \langle \mathcal{T}, \mathcal{A} \rangle$, if $\mathcal{KB} \models \vec{a}\colon \vec{Z}$, then the equivalence holds:

$$\mathcal{KB} \models \rho(\vec{b})\colon \vec{Y} \ \wedge\ \exists \vec{u}\ \Phi(\pi(\vec{a}), \rho(\vec{b}), \vec{u}) \quad \text{iff}$$
$$\mathcal{KB} \models \vec{b}\colon \vec{W} \ \wedge\ \exists \vec{v}\ \Psi(\vec{a}, \vec{b}, \vec{v}),$$

where $\pi(\vec{a})$ and $\rho(\vec{b})$ are the permutations of $\vec{a}$ and $\vec{b}$ according to π and ρ.

Intuitively, this means that, modulo some re-arrangement of the input and output vectors, the services S and Q return the same answers on any input that conforms to the request Q.

The need to permute inputs and outputs of Q to "fit" the ones of S is by no means new—it is present in any reasonable definition of service matching. Thus, in order to check whether S matches Q, a reasoning system must "guess" two appropriate permutations or exhaustively explore all possible assignments. Condition **(i)** is quite standard; for example, it can be found in definitions for matching of OWL-S services [5]. In contrast, condition **(ii)** is—to the best of our knowledge—new, and it is not expressible in terms of OWL-S service profiles.

The above definition covers only the case when S and Q have the same number of inputs and the same number of outputs. Various generalisations and extensions are presented in more detail in the technical report [1]. Therein, the reader can also find the proof of the following main statement for our approach.

Theorem. *The service matching problem w.r.t. an ontology is decidable. More precisely, it is reducible to the subsumption of conjunctive queries w.r.t. a TBox.*

The decidability and complexity of the latter problem is extensively explored for many practically interesting Description Logics (cf. [4, 7, 8, 10]; see also an overview of these results in [1]).

3 Describing and matching biomedical services

In this section, we will show the applicability of our approach to the realistic examples of web services from the biomedical domain.

3.1 Matching atomic services

Consider the following two services which extract the DNA sequence from a GenBankRecord.

```
S1: INPUT  x:GenBankRecord
    OUTPUT  y:DNASeqRepresentation
    [ hasPart(x,y) ]

S2: INPUT  x:GenBankRecord
    OUTPUT  y:DNASeqRepresentation
    THERE IS SOME d,e
   [ DNASequence(d), EMBLRecord(e),
   about(x,d), about(e,d), hasPart(e,y)]
```

They coincide on their inputs and outputs, yet they will behave in slightly different ways. The first service simply extracts the DNASequence from the input, whereas the second one first extracts the DNA sequence and then translates the syntax from GenBankForm to EMBLForm. Since there at least 20 different formats for representing DNA sequences, we have to distinguish between a DNA sequence and its representation in one of these formats.

Now consider the following request, which describes services taking a GenBankRecord and returning the corresponding DNA sequence in EMBL format:

```
Q: INPUT  x:GenBankRecord
   OUTPUT  y:DNASeqRepresentation
   THERE IS SOME d,e
[DNASequence(d), Record(e), about(x,d),
 about(e,d), hasPart(e,y), EMBLform(y)]
```

Note that, according to our definition of matching, the service `S1` does not match our request `Q` since it cannot guarantee that the output is in EMBL format. In contrast, if our TBox contains

```
SubClassOf(EMBLRecord
           restriction(hasPart
           allValuesFrom EMBLform))
```

which ensures that all entries in an EMBLRecord are in EMBLform, then service `S2` matches our request—which is indeed useful. Similarly, in the presence of the above OWL axiom, `S2` even matches the following request—despite the fact that the output of this request is more specific than that provided by the service `S2`.

```
Q1: INPUT  x:GenBankRecord
   OUTPUT  y:IntersectionOf(
        DNASeqRepresentation  EMBLform)
   THERE IS SOME d,e
  [ DNASequence(d), EMBLRecord(e),
  about(x,d), about(e,d), hasPart(e,y)]
```

3.2 Matching complex services

Next, we consider a request for a service, which takes a BlastReport and extracts its DNA sequence representation:

```
Q2: INPUT  x:BlastReport
   OUTPUT  y:DNASeqRepresentation
   [ hasPart(x,y) ]
```

This is a rather simple request, but consider the following Web Service:

```
S3: INPUT  x:BlastReport
   OUTPUT  y:DNASeqRepresentation
   THERE IS SOME p [ hasPart(x,p),
   PairWiseAlignm(p), hasPart(p,y) ]
```

If our TBox contains a statement that `hasPart` is transitive, then `S3` matches `Q2`. Similarly, if we consider the following two services, then their composition $\texttt{S4} \circ \texttt{S5}$ matches our request.

```
S4: INPUT  x:BlastReport
   OUTPUT  y:PairWiseAlignmnt
   [ hasPart(x,y) ]

S5: INPUT  x:PairWiseAlignmnt
   OUTPUT  y:DNASeqRepresentation
   [ hasPart(x,y) ]
```

In contrast, the following service does not match `Q2` because, besides extracting the sequence, it also computes its reverse complement.

```
S6: INPUT  x:BlastReport
   OUTPUT  y:DNASeqRepresentation
   THERE IS SOME p,z,w [ hasPart(x,p),
    PairWiseAlignmnt(p), hasPart(p,z),
    complementOf(z,w), reverseOf(w,y) ]
```

Let us point out again that our definition of matching yields both a higher precision and a higher recall than any comparison of inputs and outputs could possibly yield: it matches services whose inputs or outputs do not match in an obvious way (such as `Q2` and `S2` above), and it does not match services despite their in- and outputs matching (such as `S6` and `Q2`). The latter point is especially important for biomedical Web Services since many take strings as in- and outputs—and thus all services would match on the grounds of in- and outputs.

Acknowledgments

The work is supported by an EPSRC grants GR/S63168/01 and GR/S63182/01 as part of the DynamO project. For more information, please visit `http://dynamo.man.ac.uk/`.

References

[1] A. Bovykin, E. Zolin. *Describing information providing services.* Technical Report, University of Manchester 2005. `http://dynamo.man.ac.uk/publ/aaai06tr.pdf`

[2] R. D. Stevens, H. J. Tipney, C. Wroe, T. Oinn, M. Senger, P. W. Lord, C. Goble, A. Brass, and M. Tassabehji. Exploring Williams-Beuren syndrome using myGrid. *Bioinformatics*, vol. 20. 2004.

[3] L. Stein. Creating a bioinformatics nation. *Nature*, 417:119–120, 2002.

[4] Tessaris, S. *Questions and answers: reasoning and querying in Description Logic.* PhD thesis, Univ. of Manchester, 2001.

[5] T. R. Payne, M. Paolucci, and K. Sycara. Advertising and Matching DAML-S Service Descriptions. *Semantic Web Working Symposium (SWWS)*, 2001.

[6] The DAML Services Coalition. Bringing Semantics to Web Services: the OWL-S approach. In *Proc. of SWSWPC'2004.*

[7] I. Horrocks, U. Sattler, S. Tessaris, and S. Tobies. How to decide query containment under constraints using a description logic. In *Proc. of LPAR'2000.*

[8] D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of PODS'98*, 149–158.

[9] F. Baader, C. Lutz, M. Miličić, U. Sattler, and F. Wolter. Integrating description logics and action formalisms: First results. In *Proc. of AAAI'2005.*

[10] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of DL'2005.*

[11] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *J. of Web Semantics*, 1, 2003.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Distributed, high-performance earthquake deformation analysis and modelling facilitated by Discovery Net

Jian Guo Liu[1], Moustafa Ghanem[2], Vasa Curcin[2],
Christian Haselwimmer[1], Yike Guo[2], Gareth Morgan[1], Kyran Mish[3],

[1]Department of Earth Science and Engineering, Imperial College, London
[2]Department of Computing, Imperial College, London
[3]Fears Structural Engineering Laboratory, University of Oklahoma

Abstract

A Discovery Net project that has been investigating the relationship between macro and micro-scale earthquake deformational processes has successfully developed and tested a geoinformatics infrastructure for linking computationally intensive earthquake monitoring and modelling. Measurement of lateral co-seismic deformation is carried out with imageodesy algorithms running on servers at the London eScience Centre. The resultant deformation field is used to initialise geo-mechanical simulations of the earthquake deformation running on supercomputers based at the University of Oklahoma. The paper describes detail of this project, initial results of testing the workflow, and follow-on research and development.

1. Introduction

Earthquakes are important geo-hazards that can produce significant land surface deformation and ground shaking causing damage to built infrastructure and loss of life. Understanding the dynamics of earthquake deformation is therefore important for earthquake hazard assessment and insight into the effects of ground shaking on built environment.

Increasingly, grid-enabled, high performance computing is enabling geoscientists to model neotectonic and earthquake processes across a range of spatial and temporal scales (from modelling earthquake ground shaking to the tectonic evolution of fault systems). In addition, advances in satellite remote sensing, such as GPS, InSAR (Inteferometric SAR), and sub-pixel image feature matching based on local correlation (imageodesy), are providing precise geodetic measurements of surface deformation at high resolution and with a broad synoptic coverage isolating the key features.

The linkage of remote sensing derived measurements and physical models is a common feature of a variety of earth system sciences, such as meteorology and oceanography. However, until now the coupling of remote sensing measurement with computer simulation in order to investigate earthquake crustal deformation has been limited to simple analytical modelling [1] that may not effectively simulate the complexity of the earthquake system. In this paper we present an overview of the results of an e-Science applications project entitled “Bridging the Macro to Micro: a computing intensive earthquake study” (henceforth abbreviated to *M2M*). This project has developed a geoinformatics infrastructure for investigating earthquake dynamics by linking computationally intensive, remote-sensing based measurement of land surface movement with high-performance geo-mechanical simulations that model a range of physical processes as well as take into account the spatial variability in material properties in three dimensions.

2. Background

Earthquake studies that link remote sensing with model simulation are usually conducted at the macro-scale and are restricted to investigating regional patterns of deformation and their cause and effect. However, the macro phenomena of earthquakes are often composed of micro-scale displacement features and kinematic indicators (e.g. shear and extensional fracture patterns) that show the dynamic nature of the deformation. In addition, micro- and nano-scale processes along active faults are fundamental to the segmentation, initiation and halting of earthquake rupturing and may be important in accommodating the release of strain disparate from the main rupture zone. Hence, understanding and modelling the relationships between macro and micro-scale deformational phenomena may lead to a better understanding of the dynamics of earthquake deformation.

The imageodesy technique, which is based on sub-pixel local correlation is an example of macro analysis conducted within earthquake studies. It maps lateral co-seismic deformation by measuring the differences in position of corresponding image features between a pre- and post-earthquake satellite image. The technique has shown to be effective at mapping the lateral deformation associated with earthquakes, particularly in the area near an earthquake rupture [5], and complements existing geodetic methods for

measuring co-seismic ground movement (e.g. InSAR , GPS).

Finite Element Modelling (FEM) is an example of micro analysis conducted within earthquake studies. FEM is a well-established technique for determining stresses and displacements in objects and systems. In the field of earthquake engineering, FEM is used extensively by structural and geotechnical engineers to investigate the effects of earthquakes on infrastructure such as buildings, bridges, and dams. In addition, FEM-based simulations are used in the fields of structural geology, and neo-tectonics to model transient stress and deformation due to tectonics. Applying FEM to model remote sensing imageodesy results is a promising approach to investigate the stress field and faulting mechanism of macro scale earthquakes through micro scale computing simulation.

Imageodesy and FEM techniques are both compute-intensive applications that require access to specialized software components executing on high performance computing facilities. The aim of our work presented here is to enable users to integrate the use of these techniques within the same earthquake study.

The work undertaken in this project is built around the analysis of data from the Ms 8.1 Kunlun earthquake, which occurred in northern Tibet in Nov 2001. This event was the largest earthquake in China for 50 years and produced a massive surface rupture zone of over 450 km making it the longest rupture of an earthquake on land ever identified. The magnitude and nature of the earthquake along with the large area that was affected provide the perfect natural laboratory in which to investigate earthquake dynamics and test the novel geoinformatics approach that has been developed during this research project.

3. Implementation

The implementation of the geoinformatics workflow has been based on the Discovery Net informatics infrastructure [4]. Discovery Net is a grid-enabled software analytics platform developed at Imperial College London, funded through a UK e-Science Pilot Project and extended within this work. The platform is based on a service-based computing infrastructure for high throughput informatics that supports the integration and analysis of data collected from various high throughput devices.

Within Discovery Net, end-user applications are constructed within a visual authoring environment as workflows that represent the steps required for executing a particular distributed computation, and the flow of information between these tasks. Within each workflow, analysis components are treated as services or black boxes with known input and output interfaces described using web service protocols. These services can execute either on the user's own machine or make use of high performance computing resources through specialised implementations. Also, Discovery Net workflows can be published through web portals and Web/Grid services, allowing users to execute distributed computations interactively and to analyse the results of their execution using a variety of visualisation tools.

As part of the M2M project, the Discovery Net infrastructure has been extended to enable integration of the domain specific measurement and modelling services based at Imperial College and the University of Oklahoma. Processing routines, developed within Matlab, have been integrated into this workflow to enable post-processing of imageodesy output and preparation of input data for the finite element analysis software. When packaged together, this workflow provides:

- Seamless integration between measurement and modelling services.
- One-click execution with minimal user input (e.g. to define material parameters).
- The workflow has been packaged as an open web-service that has the potential to be used by the earth science community.

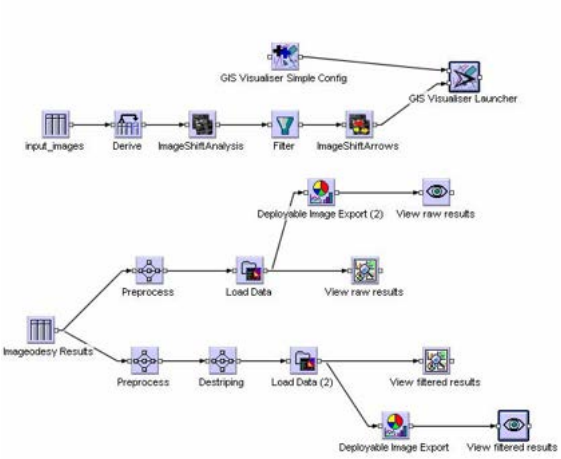


Figure 1: Imageodesy refinement workflow

The methodology that has been developed as part of the M2M project uses regional measurements of co-seismic deformation to directly run high-resolution geo-mechanical simulations of the earthquake deformation based on the Finite Element Method (FEM). Macro-scale measurements of surface deformation are retrieved using sub-pixel FNCC (Fast Normalised Cross-Correlation) imageodesy algorithms that have been developed [2] and refined [3] by the geo-application research team of Discovery Net. These algorithms measure the lateral co-seismic deformation (horizontal image feature shift and long a fault line) associated with an earthquake from pre- and post-event satellite images. The algorithms are computationally intensive and are implemented on supercomputers based at the London eScience Centre. The output deformation field produced by this software provides the input into a Finite Element analysis system, which is running on servers based at the University of

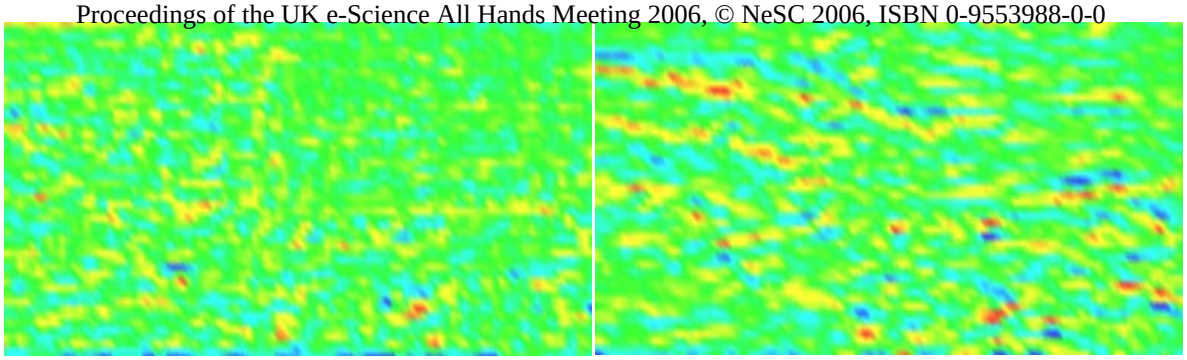


Figure 2: W-E shear strain (left) and N-S shear strain (right) during the Kunlun earthquake predicted from running finite element analysis on low-resolution imageodesy-derived co-seismic measurements using a simple 2D homogeneous elastic model. The predicted N-S shear strain field is particularly interesting as it highlights the predominantly left lateral faulting in the area.

Oklahoma. Discovery Net is used to orchestrate the domain-specific tasks (Figure 1).

The finite-element segment of the analysis was implemented on high-performance servers based at the University of Oklahoma. The *TeraScale* computational architecture [8] was used as the basis for construction of a finite-element application capable of ingesting macro-scale imageodesy measurements. The TeraScale framework was employed for this task as it provides the tools for building feature rich finite-element applications that are easily interoperable and readily scaled to the available or required computing resources. The finite-element application built during this research project enables automatic mesh creation and the initialisation of model runs using the kinematic data as boundary conditions within a full-physics geo-mechanical simulation that includes the effects of momentum conservation and material characterisation. The resultant finite-element model can be used to estimate residual stress distribution induced (or relieved) in the seismic region. In addition, macro-to-micro scale analysis is achieved by using the macro-scale kinematic measurements to define boundary conditions for higher resolution finite element models. In effect, this means that measured displacements define the boundary conditions at the edges of high-resolution finite element meshes.

4. Results

As part of the initial experiment, the Discovery Net geo-application research team has developed and applied imageodesy algorithms to cross-event Landsat 7 ETM+ satellite imagery for the Kunlun earthquake [2].

Due to the large size of the input datasets and demanding nature of the algorithms, implementation was carried out on a MPI parallel computer within the London eScience centre. The results have revealed stunning patterns of co-seismic left-lateral displacement along the Kunlun fault and presented the first 2-D measurement of regional deformation associated with this event [6]. In addition, interesting patterns of deformation south of the main fault have led to the discovery of previously unreported surface rupturing south of the main Kunlun fault [7].

We have further refined imageodesy algorithms developed as part of this pilot project [2] to include data refinement and post-processing algorithms [3] that improve the quality of the imageodesy output by removing systematic noise. The images shown on Figure 3 show the full width of the central part of the original scene. In these images, blue to green indicates shift to the left and yellow to red indicates shift to the right. The red arrows in the filtering refined image indicate the Kunlun fault zone along which the massive earthquake occurred. As the result, the terrain block south to the fault moved toward east (right). We can see how the vertical noise is very visible after the horizontal shift has been removed, while in the result produced by FFT selective and adaptive filtering; the horizontal striping and the multiple frequency wavy patterns of vertical stripes have also been successfully removed. This clearly revealed the phenomena of regional co-seismic shift along the Kunlun fault line as indicated by three arrows.

The workflow and software we have developed has been successfully tested using low-resolution imageodesy measurements to run a simple homogeneous elastic model of the seismic region (Figure 2). The macro-scale results highlight the predominantly left-lateral deformation (which ranges from 1.5-8.1m) during the Kunlun earthquake in the modelled N-S shear strain field. In addition, this result indicates the presence of unexpected but significant shear features in the northwest of the study area that are currently under further investigation.

5. Conclusions

The applications research conducted in this project has successfully developed and tested an analytical system for linking kinematic measurements of earthquake co-seismic deformation with geo-mechanical modelling to enable investigation of the dynamics of earthquake crustal deformation at macro-to-micro scales. This approach is both novel in its use of distributed high

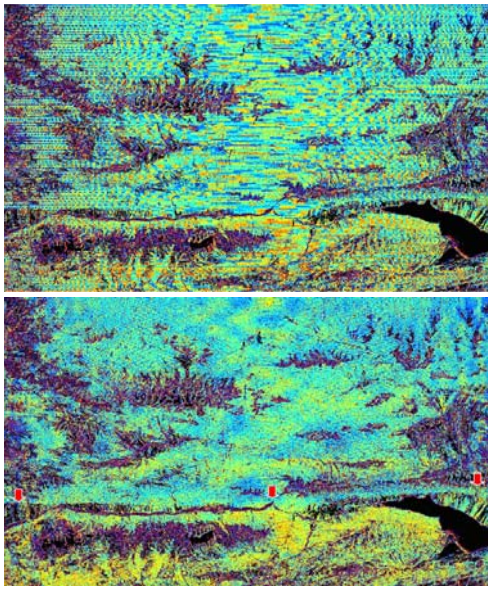


Figure 3: Filtering image noise. The original X-shift image on top is badly contaminated by stripes revealing the compensation errors of two way cross-track scanning. Using novel frequency selective and adaptive FFT filters this has been effectively removed without subduing the lateral shift information of co-seismic deformation.

performance computing and its application of advanced analytical and modelling tools to real-world problems.

With respect to geoinformatics work involved, a number of discoveries have been made:

- Novel patterns of co-seismic left-lateral displacement
- First 2D measurement of the Kunlun earthquake
- Novel rupturing south of the main fault
- Unexpected shear northwest of the main area.

One of the most important considerations when developing a valid geo-mechanical simulation is the ability to properly characterise the material properties of the region under investigation. Conventional approaches to the coupling of remote sensing measurement with physical models have been based on analytical methods that assume homogeneous geological properties for the region under investigation [1]. Our FEM approach is significantly more flexible than this and permits the construction of models containing a variety of different materials (i.e. rock types). This gives FEM the ability to construct a more realistic and heterogeneous material representation of the area under simulation. This ability is of particular importance when it comes to understanding the relationship between macro- and micro-scale deformational phenomena where differences in geological properties can be critical to the partitioning of stress and strain.

Having successfully tested our analytical workflow, our research is now focused on improving the modelling resolution and complexity in order that macro to micro-scale analysis of the Kunlun earthquake can be undertaken. The approach that we have adopted will seek to incrementally increase the modelling resolution, improve how the models are parameterised (e.g. with realistic geological properties for the region), and strive to simulate the full range of physical processes involved with earthquake deformation.

Following on from our initial low-resolution, 2D model runs we intend to run much higher resolution 2D analyses before ultimately undertaking 3D simulation of the seismic region. Parameterisation of these models will be undertaken with a more realistic characterisation of the material properties of the region. Due to the flexibility and interoperability of our finite element application the implementation of these increasingly complex simulations will be straightforward and will allow us to concentrate on interpreting the results.

References

[1] Okada, Y., 1985, Surface deformation due to shear and tensile faults in a half-space: Bulletin of the Seismological Society of America, v. 75, no. 4, p. 1135-1154.

[2] J. G. Liu and J. Ma. Imageodesy on MPI & grid for co-seismic shift study using satellite imagery In Proceedings of the 3rd UK e-Science All-hands Conference AHM 2004, Pages 232-240. Nottingham UK, September 2004.

[3] Jian Guo Liu and Gareth L. K. Morgan, "FFT Selective and Adaptive Filtering for Removal of Systematic Noise in ETM+ Imageodesy Images", accepted by the IEEE Transactions on Geoscience and Remote Sensing, 2006.

[4] S. Al Sairafi, F. S. Emmanouil, M. Ghanem, N. Giannadakis, Y. Guo, D. Kalaitzopolous, M. Osmond, A. Rowe, J. Syed and P. Wendel. The Design of Discovery Net: Towards Open Grid Services for Knowledge Discovery. International Journal of High Performance Computing Applications. Vol 17 Issue 3. 2003.

[5] Michel, R., and J.P. Avouac, Imaging Co-Seismic Fault Zone Deformation from Air Photos: The Kickapoo Stepover along the Surface Ruptures of the 1992 Landers Earthquake, J. Geophys. Res.,2006.

[6] Liu, J. G., Mason, P. J. and Ma, J., 2006. Measurement of the left-lateral displacement of Ms 8.1 Kunlun earthquake on 14th November 2001 using Landsat-7 ETM+ imagery. *International Journal of Remote Sensing*, **27**, No.10, 1875-1891.

[7] J. G. Liu, C. Haselwimmer, 2006. Co-seismic ruptures found up to 60 km south of the Kunlun fault after 14 Nov 2001, Ms 8.1, Kokoxili earthquake using Landsat-7 ETM+ imagery. *International Journal of Remote Sensing*, in press.

[8] Terascale Frameworks white paper, 2003. Available at http://nees.ou.edu/frameworks.pdf. Accessed on 4th May, 2006.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

'SED Service': Science Application based on Virtual Observatory Technology

[1]**P. Prema**,[1]N.A. Walton,[1]R.G. McMahon
[1]Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge, CB3 0HA

Abstract

We describe the development of a Spectral Energy Distribution (SED) matching technique using the current technology available through the Virtual Observatory. We outline the making of a detailed workflow technique that can take observational data of objects from various astronomical data archives and matches them to models generated through various model codes available. The result of running this workflow will produce plots of the best fit models to the data along with tabular data representing the best fit model and the closest matches, a standard error analysis on various physical parameters such as stellar mass and finally a set of image cutouts of the object. The technique will utilise various VO tools for data discovery, data access and data processing within AstroGrid, the UK's Virtual Observatory initiative.

1. Introduction

Astronomers today face significant technological challenges when handling the large data sets that are available from multi-wavelength surveys of the night sky. For example, the Sloan Digital Sky Survey (SDSS) (York et al. 2000) is producing terabytes of data every few nights. It now becomes impractical for each astronomer to have a copy of the data sets. This causes two problems, first the transportation of large (terabyte) data sets to the individual astronomers becomes impractical when considering the sheer number of people wanting the data. Secondly, even with these large data sets astronomers would spend more time reducing them rather than doing analysis on the data to produce useful scientific results. The Virtual Observatory (VO) was created to deal with these problems with two main aims. The first was to enable astronomers access to large sets of astronomical data which are stored in VO databases around the world. This is now a well established service in many VO projects around the world[1] where, for example, the user can query a database for astronomical objects through various selection criteria. The second was to provide the applications to analyse the selected data and produce useful scientific results. Thus, this project was designed to exploit the emerging VO technology to create a science application that can produce useful publishable results from the large area survey data to create large samples of objects. For example, through our technique we can determine the star formation history (SFH), star formation rates (SFR), ages and stellar masses of a large sample of high redshift galaxies based on model fits to observed galaxy photometry. So, using the data to create large samples of galaxies at various epochs along with the parameters above, astronomers can study galaxy formation and evolution problems while minimising the time spent obtaining and reducing data. We complete the introduction with a review of AstroGrid and some of its capabilities. The sections that follow describe first outline of the science and results we expect from this technique. Finally, we look at our technique from a software perspective describing the use of various AstroGrid tools.

1.1. AstroGrid

AstroGrid[2] is the UKs VO project designed to aid astronomers in research through easy data

[1]See http://www.ivoa.net

[2]http://www.astrogrid.org

access whilst also providing tools and applications for research. AstroGrid (AG) is now at a stage where it is fully developing its tools and applications that will be able to achieve useful scientific results. It is currently the only VO project to utilise a workflow system. The workflow works like a basic program where steps are executed in a logical manner. The workflow can be complex with a number of applications involved, all configured into default runtime configuration parameters.

2. Science Outline

The motivation for this 'Spectral Energy Distribution (SED) Service' is a result of the recent, e.g. the last ten years, studies of high redshift galaxies (e.g. Stanway et al. 2004, Bunker et al. 2004 and Eyles et al. 2005) discovered with the method pioneered by Steidel et al. (1995) used to identify galaxies above $z > 2$. The 'SED Service', simply put, is a matching technique that takes observational flux measurements of objects in different wave bands and matches them to model spectral energy distributions (SED) created from model codes. This process allows the user to derive physical parameters such as SFH's, stellar masses and ages from the models which can be used for analysis of the sample of objects in that given field. The technique is fast becoming a useful way to study large samples of objects at these high redshifts. Thus, creating an automated technique to study these samples of objects is very useful especially now that VO tools are in a position to support scientific analysis.

2.1. Source Extraction

Source extraction is a procedure for calculating fluxes of objects from imaging data. There are four steps to this process going from raw imaging data to a catalogue of objects with photometric redshifts. The first is to extract sources from the imaging data where we use the Source Extractor (SExtractor) photometry package (Bertin and Arnouts 1996) to create flux measurements for detected objects. Second, cross match objects detected in the different wavelength bands to form a complete catalogue of objects. Third, select a specific sample of objects based a set of colour criteria. Finally, complete the procedure by calculating photometric redshifts using, for example, Hyperz (Bolzonella et al. 2000).

2.2. Comparison with models

The information obtained from object photometry is greatly increased by matching the observational spectral energy distributions (SED) to that of synthetically constructed galaxy spectra. As an example we show the use of the Bruzual and Charlot models of 2003 (Bruzual & Charlot 2003) although other models include PEGASE (Fioc & Volmerange 1997) and Starburst99 (Leitherer et al. 1999) will shortly also be integrated. Figure 1 shows a SED match to one of the object in the GOODS[3] field. The fitting utilises the χ^2 minimisation technique to find a best fit model to the data.

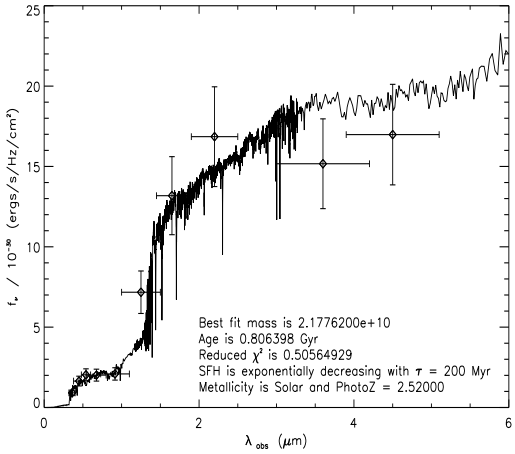


Figure 1: *The best fit GALAXEV model for our example object with an exponentially decaying SFH with the chosen e-folding timescale tau = 200 Myr and with an age of 800 Myr.*

2.3. Science Results

The scientific outcomes for such a technique will give estimates on various physical parameters including star formation rates (SFR), star formation histories (SFH), stellar masses, ages and galaxy colours. To quantify the errors on these results a standard error analysis can also done based around a χ^2 minimisation, Figure 2. This shows the confidence region for the stellar masses from the best fit model for this particular galaxy. Other outputs include tabulated data and cutout images of each object. VO tools and applications have already been put to good use by

[3] http://www.stsci.edu/science/goods/

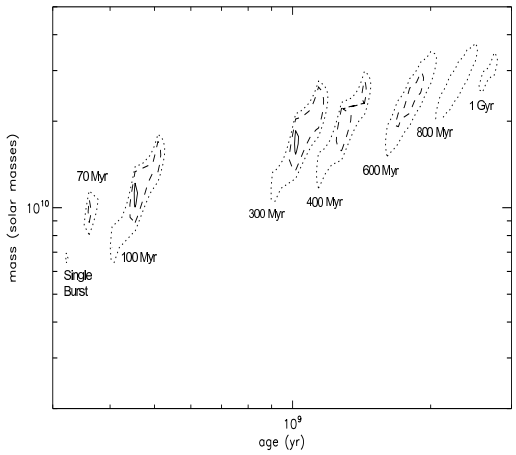


Figure 2: *Confidence levels for the stellar mass of our object for the various models shown. The contours show the 68% (solid line), 90% (dashed line), and 99% (dotted line) confidence of the chi-squared fitting.*

Padovani et al. (2004) who use VO software to discover optically faint obscured quasars.

3. Technique Description

In this section, we describe the technique from the viewpoint of the usage of AstroGrid tools. The first part of the technique is resource discovery using the AstroScope tool. This is simply the available resources for a particular region of sky. The next stage is access to the required resources which is provided by the Data Set Access (DSA) module. This works along side the Common Execution Architecture[4] (CEA) module (Harrison et al. 2005) which uses the data to run various applications specified by the user. Finally, after some data processing using various astronomy applications built into AstroGrid the data is matched to a large grid of models where a χ^2 minimisation process is used to fit the data.

3.1. Resource Discovery - AstroScope

The initial step in the process is the localisation of relevant input data sets. Use will be made of AstroScope, a relatively new tool within AstroGrid which allows the user to query various databases for imaging, catalogue and spectral data for a given region of sky. The user inputs a specific target in the sky or name of a well known object with a region they wish to explore. AstroScope then goes to various VO databases asking for information relating to the query sent by the user. The results are displayed in a tree like fashion from which the user is able to save relevant data into their MySpace area in AstroGrid.

3.2. Data Access and Retrieval

Accessing data products through AstroGrid involves two software components: DSA (data set access) and CEA (common execution architecture) modules. These modules as well as most AstroGrid components are based around web services using the SOAP[5] protocol (Rixon 2005). SOAP is a web services protocol for exchange of structured information in a decentralized, distributed environment and in AG SOAP is used in conjunction with WSDL[6] (Web Services Description Language) contracts which is a network service formatted in XML.

The underlying features that make up the DSA module are these web service features. In astronomy, most data comes in three forms: Image, Spectral and Tabular data, as shown above. Depending on the data required a query is sent using either the SIAP[7] (Simple Image Access Protocol), SSAP[8] (Simple Spectral Access Protocol) or a ADQL[9] query. For example, the first step in the 'SED Service' will have the user input a region of sky that he wishes to look at and if he chooses images the query is converted into a SIAP query that returns pointers to the data, and the same for spectral data and tabular information (catalogues).

3.3. Data Processing

Applications in AG are executed by the Job Execution System (JES) from initialisation to completion. The user will submit a query through the workbench (AG's UI) which goes through the ACR (AstroGrid Runtime Client). The ACR is a software library that allows desktop applications to call remote services, such as SExtractor, as if they

[4]http://www.ivoa.net/Documents/Notes/CEA-/CEADesignIVOANote-20050513.html

[5]http://www.w3.org/TR/soap/

[6]http://www.w3.org/TR/wsdl

[7]http://www.ivoa.net/Documents/WD/SIA/sia-20040524.html

[8]http://www.aoc.nrao.edu/∼dtody/ssa/ssa-v091.pdf

[9]http://www.ivoa.net/Documents/WD/ADQL-/ADQL-20050624.pdf

where local objects. As an example, we describe the process using the SExtractor application as represented as a first step in the 'SED Service' workflow. The ACR then talks to the JES, the AG workflow engine, since in this instance the ACR does not talk directly to the CEA but through the JES. The JES then talks to the CEA server which invokes the SExtractor application but via the CEC (Common Execution Connecter). The CEC is a WSDL contract for accepting jobs from, e.g. the JES, and allowing clients to track and retrieve results. The JES is also responsible for reading the name of the SIAP application from the workflow document sent by the user which invokes the SIAP service to retrieve the results. All data handling carried through this workflow is managed through the File Manager web application which talks to the data centre, the file store (i.e. MySpace) and the CEA server running the application. The invocation of the SExtractor application can be seen diagrammatically through figure 3. Other applications in our technique, such as the photometric redshift makers, will be executed in a similar manner.

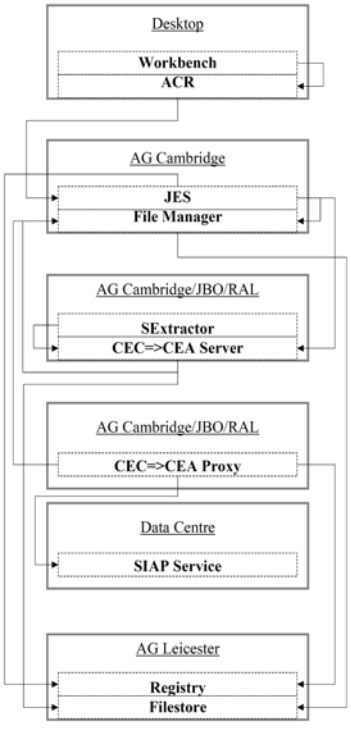


Figure 3: *The components that are engaged for the SExtractor application. A similar process is invoked when running other applications.*

4. Summary

We have highlighted how the VO can be a very useful tool for achieving publishable scientific results. Specifically taking advantage of the vast amounts of computer power currently available through AstroGrid to process the large area survey data currently being produced. The scientific analysis and results are shown to be very useful since they can estimate the physical parameters of high redshift galaxies. Thus, this leads to an increased knowledge of the formation and evolution of galaxies. We also show how this 'SED Service' can be implemented as a workflow system using AG tools and services. Design of the service is now complete and the technical implementation is now taking place for early use, summer 2006. It is anticipated that the 'SED Service' prototype will be demonstrated at the September 2006 All Hands Meeting 2006 on the PPARC eScience stand.

5. Acknowledgments

PP acknowledges support of a PPARC eScience PhD studentship.

References

[1] Bertin E., Arnouts S., 1996, AASS, 117, 393
[2] Bolzonella M., Miralles J.M., Pell R., 2000, AA, 363, 476
[3] Bruzual G., Charlot S., 2003, MNRAS, 344, 1000
[4] Bunker A.J. et al. 2004, MNRAS, 355, 374
[5] Eyles L.P. et al. 2005, MNRAS, 364, 443
[6] Fioc M., Volmerange B.R., 1997, AA, 326, 950
[7] Guy Rixon, Private Communication, 2005
[8] Harrison et al., ADASS XIV, 2005, ASP conference series, 347, 291
[9] Leitherer C. et al., 1999, AJSS, 123, 3
[10] Padovani P., Allen M.G., Rosati P., Walton N.A., 2004, AA, 424, 545
[11] Stanway E.R. et al., 2004, ApJ, 607, 704
[12] Steidel C.C., Pettini M., Hamilton D., 1995, AJ, 110, 2519
[13] York D.J. et al., 2000, AJ, 120, 1579

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Topology-aware Fault-Tolerance in Service-Oriented Grids

Paul Townend, Jie Xu
School of Computing,
University of Leeds,
Leeds, UK

{pt,jxu}@comp.leeds.ac.uk

Abstract

A promising means of attaining dependability improvements within service-oriented Grid environments is through the set of methods comprising design-diversity software fault-tolerance. However, applying such methods in a Grid environment requires the resolution of a new problem not faced in traditional systems, namely the common service problem whereby multiple disparate but functionally-equivalent services may share a common service as part of their respective workflows, thus decreasing their independence and increasing the likelihood of a potentially disastrous common-mode failure occurring. By establishing awareness of the topology of each channel/alternate within a design-diversity fault-tolerance scheme, techniques can be employed to avoid the common-service problem and achieve finer-grained control within the voting process. This paper explores a number of different techniques for obtaining topological information, and identifies the derivation of topological data from provenance information to be a particularly attractive technique. The paper concludes by detailing some very encouraging progress with integrating both provenance and topology-aware fault-tolerance applications into the Chinese CROWN Grid middleware system.

1. Introduction

A traditional way to increase the dependability of distributed systems is through the use of fault tolerant techniques [AND90]. The approach of design diversity - and especially multi-version design - lends itself to service-oriented architectures (SOAs) and hence Grids, as the potential availability of multiple functionally-equivalent services should allow a multi-version system to be dynamically created at much lower cost than would traditionally be the case. This is because a developer can pay to license/use pre-existing services rather than develop and maintain their own. Service-orientation promises to reduce the cost of developing and maintaining any in-house services, as well as the cost of integrating multiple services together [LIN03].

However, the provision of design-diversity fault-tolerance in Grid environments brings with it a new problem not encountered in traditional systems, whereby multiple disparate but functionally-equivalent services may – dynamically or statically during the course of their workflow – invoke identical 'common' services, thus reducing channel diversity, and potentially increasing the likelihood of common-mode failure occurring (whereby multiple channels fail in a similar way, producing identical but incorrect results) [TOW05a]. We refer to this problem as the *common-service problem.* A potential solution to this problem is through the analysis of topological information of constituent channels/alternates within a design-diversity system. Methods of deriving such topological information are analysed within this paper, with the use of provenance information shown to be particularly effective. Furthermore, this paper demonstrating the successful integration of an existing provenance Grid system to an existing fault-tolerance Grid system, in order to allow for topology-aware fault-tolerance in a Grid environment. The feasibility of topological awareness is further enhanced by the integration of this system into the CROWN Chinese Grid middleware system.

2. Topological analysis

Services which are composed of other services are known as *composite services*, whereas services which only access their local system are known as *basic services.* Whether or not a service is composite or not is usually hidden from the perspective of a service requestor; indeed, the distinction between a basic service and a composite service is often seen as unimportant; for example, in the context of Web services, [ALO04] states "*…whether a Web service is basic or composite is irrelevant from the perspective of the clients, as it is only an implementation issue.*"

However, the concept of a service being potentially composite takes on much greater importance when considering the common-service problem, due to the lack of knowledge about the workflow of alternates/channels within a design diversity scheme. This results in design diversity becoming less attractive in the context of SOAs, despite the great opportunities for cheap construction of such systems. It is therefore essential to investigate ways in which to reduce the impact of such common services between channel/alternate workflows. One such method is to exploit awareness in system topology to achieve this goal, as through knowledge of system topology it may be possible to devise techniques for reducing the impact of the common service problem, such as through the use of weighted voting algorithms.

The topology of a software system is essentially the set of dependencies between the components that constitute that system. In the context of the common service problem, it is important that the dynamic dependencies that can occur due to the ultra-late binding capability of service-oriented Grids are properly represented within the topological view of the system, and therefore we are particularly concerned with building up a view of system topology that represents the concrete workflows of every channel/alternate within a design diversity based system that are enacted for a given input, and – recursively - the concrete workflow of each component within those workflows.

We define topology as "*the set of dependencies between all components of a system that are used to produce an outcome based on a given input*", where a system consists of a number of components, which cooperate under the control of a design to service the demands of the system environment. In order to provide such topology information, there needs to be mechanisms to extract it.

2.1 Methods of achieving topological awareness

At the current moment in time, there are no widely known methods of gaining information about the topology of a given service within a Grid system, and very little research has been performed in this area, the notable exception being that performed by [PER05]. However, there are a number of technologies that can potentially be used to provide such information.

One such technology is that of SOAP messaging. SOAP is the commonly used XML-based protocol used when passing messages between services within Grids. The body element of a SOAP message is concerned with the main information that the sender wishes to transmit to the receiver, whilst the header element (if present) contains additional information necessary for intermediate processing or added value services. One possible method of recording topology information is to require the hosting environment of each service within a Grid to append topology information within the header of every outgoing SOAP message it sends. In this way, the topology of a given request can be reconstructed from the individual blocks within the SOAP header element, once the final SOAP message containing a result is received by the fault tolerance mechanism. This is a similar to the approach used in [PER05], which attempts to gather topological information by prepending dependency information onto a CORBA request as it passes from stage-to-stage.

Another potential means for storing topological information is through extensions to WSDL documentation. A WSDL document describes the interface of a Web service and how to interact with that interface; however, an extension to WSDL could allow for information about the workflow of a given service/operation to be expressed within the service description of a service. A service requestor could thus parse this information in order to ascertain the workflow of a given service operation.

A similar technique to this is to store information about the topology of a given service's workflow (or alternate negotiable topologies) in a metadata document (such as a *WS-Policy* document) related to that service; this data can then be retried via a technology such as *WS-MetadataExchange* and parsed accordingly.

The advantage and disadvantages of each of these techniques are summarised below:

• *Appending topology information to SOAP headers.* This method shows promise, as it allows for the capture of dynamic system topology; this is a particularly useful property in the context of service-oriented Grids, due to their ability to enact late-binding schemes. However, the method also requires that each hosting environment within a virtual organisation be adapted to perform this task in a syntactically and semantically equivalent way; as there is currently no existing technology that provides this functionality, this is a non-trivial task. Also, the approach could – depending on the size of a workflow (which may be recursive) – result in very large SOAP headers that require increased time to parse and transmit, thus resulting in increased system overhead.

• *WSDL*. Embedding topological information within the WSDL document of a service has the advantage that topology can be derived before the service is invoked; this results in increased choice for any topology-aware fault tolerance mechanisms as decisions can be made ahead of invocation. However, this method cannot express the topology of a service that exploits dynamic, ultra-late binding techniques, and requires that any fault tolerance

mechanism that wishes to exploit topological data to always have access to the latest version of the service's WSDL document. Additionally, such a method would require extensions to the standard WSDL schema that currently do not exist, and would have to be adhered to by each service within a workflow recursively.

• *Metadata*. The notion of embedding topological information within the metadata of a service has similar advantages to that of embedding the information within a WSDL document; namely that topology can be extracted in advance of the invocation stage. However, the approach shares a similar disadvantage in that dynamic ultra-late binding services (in any part of the set of workflows enacted) cannot be modelled in this way. Unlike embedding topological information within a WSDL document, however, there are already standard ways – such as WS-Policy and WS-MetadataExchange – for expressing and extracting metadata, and so at least from a syntactic point of a view, the approach is more feasible.

2.2. Use of Provenance

In addition to the methods expressed above, we propose a new technique to deriving topological information about service workflows: analysis of the provenance information of a given task. The provenance of a piece of data is the documentation of the process that led to a given data element's creation. This concept is well established, although past research has referred to the same concept with a variety of different names (such as lineage [LAN91], and dataset dependence [ALO97]).

The recording of provenance information can be used for a number of goals, including verifying a process, reproduction of a process, and providing context to a piece of result data. Provenance in relationship to workflow enactment and SOAs is discussed in [SZO03]; in a workflow based SOA interaction, provenance provides a record of the invocations of all the services that are used in a given workflow, including the input and output data of the various invoked services. Through an analysis of interaction provenance, patterns in workflow execution can thus be detected. Using actor provenance, further information about a service, such the script a service was running, the service's configuration information, or the service's QoS metadata, can be obtained. Through an analysis of interaction provenance, system topology can thus be derived.

This approach to deriving topological information has a number of advantages over all three of the methods described in section 2.1. Methods of deriving topology from WSDL descriptions or service metadata are attractive in that the entire topology of a process can be known ahead of invocation (prospectively); however, these schemes assume a static topology and thus cannot be used with dynamic, ultra-late binding services. Conversely, recording topology within SOAP header elements can be used to record dynamic topology, but is strictly retrospective – the topology of a process can only be analysed upon the return of a result.

By deriving topology information from a provenance system, a neat compromise can be made between these properties; topology information can be derived at run-time during the course of an invocation (and subsequent workflow enactment) by querying data contained within the provenance store; this allows some analysis to be performed before a result is received, and also supported dynamic, ultra-late binding services.

Additionally, provenance technologies are already in existence, and are being increasingly used in real-world hosting environments, and so the infrastructure on which topology information can be extracted does not need creating, unlike in the case of appending information to SOAP messages.

Table 1. Methods for topology derivation.

Scheme	Dynamic	Data Availability	Feasibility
SOAP headers	Yes	Retrospective	Requires implementation of middleware support.
WSDL	No	Prospective	Requires extensions to specification.
Metadata	No	Prospective	Technology already exists.
Provenance	Yes	Run-time.	Technology already exists.

A comparison of all four schemes is shown in table 1. Due to the advantages it provides over other possible methods, the view of this paper is that provenance is a particularly attractive technique to consider when deriving topological information.

3. PreServ Integration with CROWN

The ideas of exploiting topological information to aid in the provision of design-diversity software fault-tolerance in Grid environments have been explored in the FT-Grid tool, developed at the University of Leeds and discussed in [TOW05b]. This tool derives its topological data from analysis of provenance records generated by the PreServ provenance recording tool [GRO04], developed at the University of Southampton. Initial experimentation has been

very positive, and has demonstrated a significant reduction in the number of common-mode failures encountered during a large number of automated tests. [TOW05b].

One drawback of the approach is the assumption that every service within a Grid system has a compatible provenance recording facility; this is a rather strong assumption, and has been of questionable feasibility.

However, recently, as part of the UK-Sino COLAB (Collaboration between Leeds and Beihang universities) project, the developers of the CROWN (Chinese Research environment over Wide-area Network – a middleware widely used in the Chinese e-Science programme) Grid middleware system have integrated the PreServ provenance system into the main CROWN middleware.

The intention of this is to allow provenance recording functionality to be transparently available on any service hosted on CROWN systems. From the perspective of using topological information to assist design-diversity fault-tolerance schemes, this is of great value, as the prospect of deriving complete topological information about channel/alternate workflows becomes much more feasible on systems developed within the CROWN Grid. Indeed, it is intention of the COLAB project to integrate a generalised FT-Grid application into future releases of the CROWN middleware, to take advantage of this situation.

This generalised FT-Grid application is intended to offer mechanisms for topological analysis that can then be fed into other fault-tolerance schemes, as well as offering to enact user-defined design-diversity fault-tolerance applications itself (such as Distributed Recovery Blocks and Multi-Version Design).

4. Conclusions

There is a great need for methods for improving the dependability of applications within service-oriented Grid environments. A promising means of attaining such dependability improvements is through the set of methods comprising design-diversity software fault-tolerance, such as Multi-Version Design and Recovery Blocks.

However, applying such methods in a Grid environment requires the resolution of a new problem not faced in traditional systems. This paper investigates the *common service problem*, whereby multiple disparate but functionally-equivalent services may share a common service as part of their respective workflows, thus decreasing their independence and increasing the likelihood of a potentially disastrous common-mode failure occurring.

By establishing awareness of the topology of each channel/alternate within a design-diversity fault-tolerance scheme, techniques can be employed to avoid the common-service problem and achieve finer-grained control within the voting process.

This paper has explored a number of different techniques for obtaining topological information, and has identified the derivation of topological data from provenance information to be a particularly attractive technique.

The paper concludes by detailing some very encouraging progress with integrating both provenance and topology-aware fault-tolerance applications into the Chinese CROWN Grid middleware system.

5. References

[ALO97] G. Alonso, C. Hagen, “Geo-opera: Workflow Concepts for Spatial Processes”, in Proceedings of the 5th International Symposium on Spatial Databases, Berlin, Germany, June 1997.

[ALO04] G. Alonso, F. Casati, H. Kuno, V. Machiraju, “Web Services: Concepts, Architectures and Applications”, Springer, 2004.

[AND90] T. Anderson and P. Lee, Fault Tolerance: Principles and Practice. New York: Springer-Verlag, 1990.

[GRO04] P. Groth, M. Luck, L. Moreau, “A protocol for recording provenance in service-oriented grids”, in Proceedings of the 8th International Conference on Principles of Distributed Systems (OPODIS'04), Grenoble, France, December 2004.

[LAN91] D. Lanter, “Lineage in Gis: The Problem and a Solution”, Technical Report 90-6, National Center for Geographic Information and Analysis, UCSB, Santa Barbara, CA, 1991.

[LIN03] Z. Lin, H. Zhao, and S. Ramanathan, "Pricing Web Services for Optimizing Resource Allocation - An Implementation Scheme," presented at 2nd Workshop on e-Business, Seattle, December 2003.

[PER05] S. Pertet, P. Narasimhan, “Handling Propagating Faults: The Case for Topology-Aware Fault-Recovery”, in DSN Workshop on Hot Topics in System Dependability, Yokohama, Japan, June 2005.

[SZO03] M. Szomszor, L. Moreau, “Recording and reasoning over data provenance in web and grid services”, Int. Conf. on Ontologies, Databases and Applications of Semantics, Vol. 2888 of Lecture Notes in Computer Science, pp. 603-620, Catania, Italy, November 2003.

[TOW05a] P. Townend, P. Groth, J. Xu, "A Provenance-Aware Weighted Fault Tolerance Scheme for Service-Based Applications", in Proceedings of 8th IEEE International Symposium on Object-oriented Real-time distributed Computing, Seattle, May 2005

[TOW05b] P. Townend, P. Groth, N. Looker, J. Xu, "FT-Grid: A Fault-Tolerance System for e-Science", in Proceedings of 4th U.K. e-Science All-Hands Meeting, 19th - 22nd Sept., 2005, ISBN 1-904425-53-4.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Alternative Interfaces for OWL Ontologies

Alan L Rector, Nick Drummond, Matthew Horridge, Hai H. Wang and Julian Seidenberg

Department of Computer Science, University of Manchester, M13 9PL UK

Abstract

This poster presents a selection of ontology editing tools that have been developed for Protégé-OWL. The tools were developed in the context of the CO-ODE project[1]. Although Protégé-OWL[2] is arguably the most widely used tool for editing OWL ontologies, the design and implementation of the standard Protégé-OWL user interface has been focused around satisfying the needs of logicians and knowledge modellers. The tools that are presented in this poster are designed to be used by domain experts who are generally not logic savvy. It is hoped that these tools will encourage e-Scientists to take up OWL for use in their service architectures.

Ontologies and metadata are key to knowledge management and service architectures for e-Science. They are critical for annotating resources so that they can be discovered and used by semantic middle-ware in service oriented architectures and workflows. In particular, the Web Ontology Language, OWL, is on the verge of gaining acceptance from a wide range of e-Science groups and organisations.

Despite the fact that OWL is generally seen as a solution to a number of design and implementation areas in e-Science, the complexity of the language, and understanding its logical underpinnings can be a barrier to its correct use. To address this situation, various ontology editors such as Protégé-OWL [1] and SWOOP [2] have been developed. However, the default interfaces in both of these tools are somewhat geared towards the tastes of logicians. These kinds of users are typically seduced by the ability to take advantage of the full expressivity of OWL. In contrast, for many other types of users, it is frequently the case that a particular application or style dictates that only a subset of the full expressive power of a language such as OWL is required. This is particularly true when the aim is to extend a simple terminology or taxonomy. In addition, most users need an environment that makes ontology development a fast and pain free process. They also require tools support for frequently used ontology design patterns, so that such patterns can be applied quickly and without error.

Using real requirements from groups, and exposure to difficulties experienced by beginners during tutorials, we have developed and are experimenting with several different user interfaces. These user interfaces attempt to address the following points:

1) Hiding the logical constructs and complex notation from the user.
2) Selecting appropriate expressiveness for a given group.
3) Abstracting away from the language by simulating some higher-level constructs.

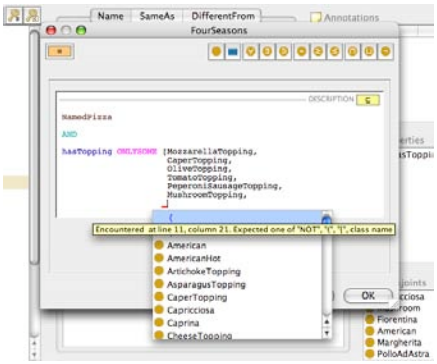
[1] http://www.co-ode.org

[2] http://protege.stanford.edu/plugins/owl

We have developed a range of plugins for Protege-OWL that help achieve the above goals. A selection of these plugins are described below.

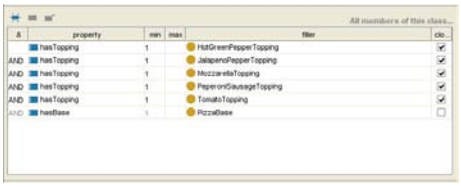
We have developed a new syntax - the Manchester OWL Syntax - for the representation of ontology classes. We have found this to be the syntax of choice for domain experts and non-logicians. In addition to the development of this syntax, a special editor has been developed (Figure 1). As well as the customary auto completion, keyword highlighting, error highlighting and pretty printing, the editor provides "macros" or shortcuts for common patterns. For example, multiple existential or universal restrictions along a particular property, coupled with closure, can be created in a very compact form. We have found that the editor results in a significant increase in the speed at which complex or long class descriptions can be constructed.

Figure 1 - The Manchester OWL Syntax Editor



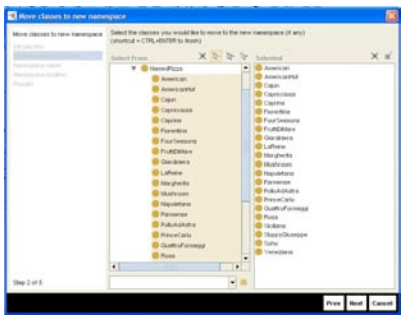
Work with the International Organisation for Terminology in Anaesthesia (IOTA) group, who are developing a medical terminology, has produced a simplified class editing interface. The interface (Figure 2) trades expressivity for simplicity. Despite this seeming reduction in expressivity, we have found it to be very popular as it allows users to enter class descriptions in a “property oriented” or UML like manner.

Figure 2 - The Simple Conditions Widget



Finally, we have developed a suite of wizards (Figure 3) that allow users to quickly and accurately complete time consuming and error prone tasks. Such tasks range from quickly inputting/sketching a class hierarchy, to filling in restrictions for multiple classes. The wizards also support several of the W3C Semantic Web Best Practices Working Group recommended ontology design patters.

Figure 3 - An example ontology wizard



Conclusion

It is hoped that the availability of the above tools will aid ontology development in e-Science projects. We are interested in using this opportunity to showcase the tools, and discuss the difficulties that users have had with developing ontologies. In particular, we would like to obtain tools requirements from researchers involved with e-Science projects.

Acknowledgements

The CO-ODE project is funded by JISC.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Constructing Chained Molecular Dynamics Simulations of HIV-1 Protease Using the Application Hosting Environment

P. V. Coveney, **S. K. Sadiq**, R. S. Saksena, and S. J. Zasada

Centre for Computational Science, Department of Chemistry, University College London, Christopher Ingold Laboratories, 20 Gordon Street, London, WC1H 0AJ

Abstract

Many crystal structures of HIV-1 protease exist, but the number of clinically interesting drug resistant mutational patterns is far larger than the available crystal structures. Mutational protocols convert one protease sequence with available crystal structure into another that diverges by a small number of mutations. It is important that such mutational algorithms are followed by suitable multi-step equilibration protocols, e.g. using chained molecular dynamics simulations, to ensure that the desired mutant structure is an accurate representation. Previously these have been difficult to perform on computational grids due to the need to keep track of large numbers of simulations. Here we present a simple way to construct a chained MD simulation using the Application Hosting Environment.

I Introduction

Computational grids [1, 2] provide an ideal environment to perform compute intensive tasks such as molecular dynamic simulations, but many scientists have been deterred from using them due to the perceived difficulty of using the grid middleware [3]. The Application Hosting Environment [4] is a lightweight, WSRF [5] compliant middleware system designed to allow a scientist to easily run applications on remote grid resources. We have successfully used it to host the NAMD [6] molecular dynamics code, and run jobs on both the UK National Grid Service and the US TeraGrid. Here we present a case study using the AHE to construct chained application workflows in an investigation into the HIV-1 protease.

II Molecular Dynamics of HIV-1 Protease

Our case study is on the use of the AHE to manage molecular dynamics simulations of the HIV-1 protease. The protease encoded by HIV is responsible for the cleavage of viral polyprotein precursors and subsequent maturation of the virus. The protease is a symmetric dimer (each monomer has 99 amino acids) that encloses a pair of catalytic aspartic acid residues in the active site. The active site is bound by a pair of highly flexible flaps that allow the substrate access to the aspartic acid dyad [9, 10].

The enzyme has been a key target for antiretroviral inhibitors and an example of structure assisted drug design [11]. Unfortunately, therapy is limited by the emergence and proliferation of drug resistant mutations in various enzymes of HIV [12]. HIV-1 protease, also exhibits tolerance to a significant quantity of non-drug resistant mutations as part of its natural variability [13]. Comparisons of resolved crystal structures of HIV-1 protease supports the stability of tertiary structure to many mutations [14].

Although many such crystal structures of HIV-1 protease exist, the scope and extent of both clinically interesting drug resistant mutational patterns [15] and non-drug resistant mutations is far larger than available by crystallographic methods. It is therefore necessary when modeling HIV-1 protease mutants to employ mutational protocols that convert one protease sequence with available crystal structure into another that diverges by a small number of mutations, but which has no crystal structure. It is also important that such mutational algorithms are followed by suitable multi-step equilibration protocols to ensure that the desired mutant structure is an accurate representation of the actual structure. Whilst standard protocols exist that employ several steps including gentle annealing to physiologically relevant temperatures, removal of force constraints on the protease and establishing a relevant thermodynamic ensemble [10], more extensive protocols are required to cope with the implementation of divergent mutations from a crystal structure.

Here we present an equilibration protocol composed of a chained sequence of molecular dynamics simulations that implements standard protocol requirements as well as including steps that allow for conformational sampling and re-

Eq Step	Procedure	Sim Duration (ps)	Force Constant (kcal/mol)	Constrained Atoms
eq0	minimization	2000 iterations	1	A
eq1	annealing: 50K - 100K	10	1	A
eq2	annealing: 100K - 300K	20	1	A
eq3*	NVT**	200	1	A
eq4	NVT	50	0.8	A
eq5	NVT	50	0.6	A
eq6	NVT	50	0.4	A
eq7	NVT	50	0.2	A
eq8	NVT	50	0.2	B
eq9	NVT	50	0.2	C
eq10	NVT	470	0	-
eq11	NPT***	1000	0	-

A = all non-hydrogen protease atoms

B = class 'A' except atoms of all amino acids within 5 Å of and including N25D mutations

C = class 'A' except atoms of all amino acids within 5 Å of and including I84V mutations

* This step prevents premature flap collapse [7]

** NVT ensemble temperature maintained using Langevin thermostat with coupling coefficient of 5 /ps

*** NPT ensemble maintained using Berendsen Barostat [8] at 1bar and with pressure coupling of 0.1 ps

Table 1: Equilibration protocol for molecular dynamics simulation of HIV-1 protease incorporating relaxation of mutated amino acid residues.

laxation of the incorporated mutations within the framework of their surrounding protease structure. Furthermore, we show how use of the AHE both automates such a chained protocol and facilitates deployment of such simulations across distributed grid resources.

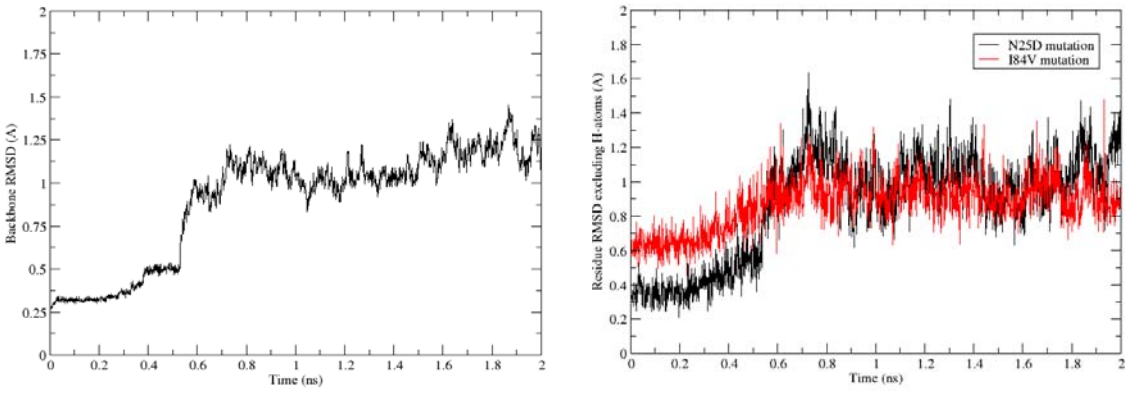
III The Application Hosting Environment

The Application Hosting Environment (AHE) is a lightweight, WSRF [5] compliant, web services based environment for hosting unmodified scientific applications on the grid. The AHE is designed to allow scientists to quickly and easily run unmodified, legacy applications on grid resources, manage the transfer of files to and from the grid resource and monitor the status of the application. The philosophy of the AHE is based on the fact that very often a group of researchers will all want to access the same application, but not all of them will possess the skill or inclination to install the application on a remote grid resource. In the AHE, an expert user installs the application and configures the AHE server, so that all participating users can share the same application. For a discussion of the architecture and implementation of the AHE, see [4] .

The AHE provides users with both GUI and command line clients to interact with a hosted application. In order to run an application using the AHE command line clients, firstly the user must issue the ahe-listapps command to find the end point of the application factory of the application she wants to run. Next she issues the ahe-prepare command to create a new WS-Resource to manage the state of her application instance. Finally she issues the ahe-start command, which will parse her application configuration file to find any input files that need to be staged to the remote grid resource, stage the necessary files, and launch the application. The user can then use the ahe-monitor command to check on the progress of her application and, once complete, the ahe-getoutput command to stage the output files back to her local machine. By calling these simple commands from a shell or Perl script the user is able to create complex application workflows, starting one application execution using the output files from a previous run.

IV Implementation

The 1TSU crystal structure was used as the starting point for the molecular dynamics equilibration protocol. This structure contains inactive wildtype protease complexed to a substrate. VMD [16] was used for the initial preparation of the system prior to simulation. The coordinates of the substrate were removed from the structure, all missing hydrogen atoms were inserted and the structure was solvated and neutralized. The N25D mutation was incorporated to restore catalytic activity to the protease and the I84V as it is a primary drug resistant mutation for sev-



(a) (b)

Figure 1: Root-mean-squared-deviation (RMSD) of protease amino acid backbone atoms excluding hydrogen, with respect to the initial X-ray structure (a) and of the dimeric pair of mutated amino acid atoms excluding hydrogen, with respect to the initial X-ray structure (b).

eral inhibitors. The molecular dynamics package NAMD2 [6] was used for all equilibration simulations.

The equilibration protocol was adapted from Perryman *et al.* [10] with several important modifications and is presented in Table 1. NAMD configuration files corresponding to each step of the equilibration protocol were set up in a way such that the output files of each step would serve as the input files of the next step. The files were generated automatically using a Perl script designed to set up such systems, and a naming convention was used for the NAMD configuration file at each step of the equilibration protocol to ease scripting of the workflow.

A Perl script was created to execute the desired equilibration chain on a remote grid resource, using the AHE middleware to manage the state of each of the steps in the chain. The script executed the ahe-prepare command followed by the ahe-start command sequentially for each step of the equilibration protocol. This had the effect of preparing a WS-Resource to manage the step, staging input files necessary for the step, and executing the application. The script then polled the AHE server at regular intervals using the ahe-monitor command until the simulation step had completed. Once complete, the script staged the files back to the local machine and used them initiate the next step of the equilibration protocol. The script terminated after sequentially executing all desired steps in the chained protocol.

V Results & Conclusion

Analysis of the root-mean-square-deviation (RMSD) of the backbone atoms of HIV-1 protease was performed (Figure 1 (a)). A slow relaxation of the backbone occurs across the first 0.4 ns of equilibration due to the gradual reduction of the force constant from 1 kcal/mol to 0.2 kcal/mol. The RMSD plateau of 0.5 Å between 0.4 ns and 0.6 ns is a signature of that part of the equilibration protocol where the force constant is maintained at 0.2 kcal/mol for most of the protease and stepwise relaxation of the mutated amino acid positions and local environment is allowed. As soon as the force constant is removed there is a rapid rise in the RMSD to approximately 1 Å away from X-ray structure as all amino acid positions along the backbone move towards optimal conformations. The mean RMSD across the last 1ns of equilibration is 1.11±0.11 Å and describes equilibration of the protease at a relatively low distance from initial X-ray structure with small fluctuations.

The RMSD of the backbone and sidechain atoms (excluding hydrogen) of the mutated amino acids was also calculated (Figure 1 (b)). The positions of both residues change relatively little during the period in which their force constants are set to zero. Once the whole protease is free from constraints, residue 25 describes an abrupt change in RMSD to approximately 1 Å whilst residue 84 changes more gradually to the same value. This may be due to the fact that although both sets of residues are in the active site, the D25 dyad is more exposed to water than V84 and thus more prone to moving once constraints have been lifted. The mean RMSD during the last 1 ns of simulation is 1.02±0.15 Å and 0.92±0.10 Å for D25 and V84 residues respectively, which is smaller than the backbone RMSD of the whole protease.

The simulation has shown that in this case, the change in RMSD of mutated amino acids is similar to that of the protease backbone as a whole. Whilst this is indicative of a good initial mutational protocol, such as that used in VMD, differences in the RMSD of mutated residues during minimization and force relaxation show that an equilibration protocol that allows conformational change of mutated amino acids assists in the achievement of equilibration. Furthermore, as a significant degree of simulation using a multi-step protocol is necessary to achieve equilibration, the ability to automate implementation of such a protocol using the AHE is greatly beneficial when considering the need to do such equilibrations for a large number of protease mutations.

We have also shown that due to the flexible nature of the AHE, a complex workflow can be orchestrated by scripting the AHE command line clients; in this case we have conducted a chained molecular dynamic simulation using less than forty lines of Perl code.

References

[1] P. V. Coveney, editor. *Scientific Grid Computing*. Phil. Trans. R. Soc. A, 2005.

[2] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Intl J. Supercomputer Applications*, 15:3–23, 2001.

[3] J. Chin and P. V. Coveney. Towards tractable toolkits for the grid: a plea for lightweight, useable middleware. Technical report, UK e-Science Technical Report UKeS-2004-01, 2004. http://nesc.ac.uk/technical_papers/UKeS-2004-01.pdf.

[4] P. V. Coveney, S. K. Sadiq, R. S. Saksena, M. Thyveetil, S. J. Zasada, M. Mc Keown, and S. Pickles. A lightweight application hosting environment for grid computing. 5th UK e-Science All Hands Meeting, 2006.

[5] S. Graham, A. Karmarkar, J Mischkinsky, I. Robinson, and I. Sedukin. Web Services Resource Framework. Technical report, OASIS Technical Report, 2006. http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-os.pdf.

[6] L. Kale, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulten. NAMD2: Greater scalability for parallel molecular dynamics. *J. Comp. Phys.*, 151:283–312, 1999.

[7] K. L. Meagher and H. A. Carlson. Solvation Influences Flap Collapse in HIV-1 Protease. *Proteins: Struct. Funct. Bioinf.*, 58:119–125, 2005.

[8] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, 81:3684–3690, 1984.

[9] W. R. P. Scott and C. A. Schiffer. Curling of Flap Tips in HIV-1 Protease as a Mechanism for Substrate Entry and Tolerance of Drug Resistance. *Structure*, 8:1259–1265, 2000.

[10] A. L. Perryman, J. Lin, and J. A. McCammon. HIV-1 protease molecular dynamics of a wild-type and of the V82F/I84V mutant: Possible contributions to drug resistance and a potential new target site for drugs. *Protein Sci.*, 13:1108–1123, 2004.

[11] A. Wlodawer and J. Vondrasek. Inhibitors of HIV-1 Protease: A Major Success of Structure-Assissted Drug Design. *Annu. Rev. Biophys. Biomol. Struct.*, 27:249–284, 1998.

[12] V. A. Johnson, F. Brun-Vezinet, B. Clotet, B. Conway, D. R. Kuritzkes, D. Pillay, J. Schapiro, A. Telenti, and D. Richman. Update of the Drug Resistance Mutations in HIV-1: 2005. *Int. AIDS Soc. - USA*, 13:51–57, 2005.

[13] N. G. Hoffman, C. A. Schiffer, and R. Swanstrom. Covariation of amino acid positions in hiv-1 protease. *Virology*, 314:536–548, 2003.

[14] V. Zoete, O. Michielin, and M. Karplus. Relation between Sequence and Structure of HIV-1 Protease Inhibitor Complexes: A Model System for the Analysis of Protein Flexibility. *J. Mol. Biol.*, 315:21–52, 2002.

[15] T. D. Wu, C. A. Schiffer, M. Gonzales, J. Taylor, R. Kantor, S. Chou, D. Israelski, A. R. Zolopa, W. J. Fessel, and R. W. Shafer. Mutation Patterns and Structural Correlates in Human Immunodeficiency Virus Type 1 Protease following Different Protease Inhibitor Treatments. *J. Virol.*, 77:4836–4847, 2003.

[16] W. Humphrey, A. Dalke, and K. Schulten. VMD - Visual Molecular Dynamics. *J. Mol. Graph.*, 14:33–38, 1996.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

PRATA: A System for XML Publishing, Integration and View Maintenance

Gao Cong Wenfei Fan* Xibei Jia **Shuai Ma**
University of Edinburgh
{gao.cong@,wenfei@inf.,x.jia@sms.,sma1@inf.}ed.ac.uk

Abstract

We present PRATA*, a system that supports the following in a uniform framework: (a)* XML *publishing,* i.e., *converting data from databases to an* XML *document, (b)* XML *integration,* i.e., *extracting data from multiple, distributed databases, and integrating the data into a single* XML *document, and (c) incremental maintenance of published or integrated* XML *data (view),* i.e., *in response to changes to the source databases, efficiently propagating the source changes to the* XML *view by computing the corresponding* XML *changes. A salient feature of the system is that publishing, integration and view maintenance are* schema-directed*: they are conducted strictly following a user-specified (possibly recursive and complex)* XML *schema, and guarantee that the generated or modified* XML *document conforms to the predefined schema. We discuss techniques underlying* PRATA *and report the current status of the system.*

1 Introduction

It is increasingly common that scientists want to integrate and publish their data in XML. As an example, consider how biologists exchange their data (the story is the same in other areas such as astronomy, earth sciences and neuroinformatics.) A biologist may manage a collection of experimental data by using some database management system (DBMS). In addition, information will be brought in from other databases and integrated with the basic data. To share information, the community of biologists interested in the topic get together and decide that there should be some standard format for data exchange, typically XML. In addition, they produce some XML DTD or schema to describe that format such that all members of the community exchange their data in XML with respect to the schema. Now one needs to handle the migration of data through various formats and to ensure that the resulting XML data conforms to the predefined schema. This is known as *schema-directed publishing/integration*. More specifically, XML publishing is to extract data from a traditional database, and construct an XML document that conforms to a *predefined* XML *schema*. XML integration is to extract data from *multiple, distributed* data sources, and construct an XML document (referred to as XML view) that conforms to a given schema.

With XML publishing/integration also comes the need for *maintaining* the published XML data (view). Biologists constantly update their databases. To propagate the source changes to the XML view, a naive approach might be to redo the publishing and integration from scratch. However, when the source data is large, the publishing and integration process may take hours or even days to complete. A better idea is by means of *incremental* XML *view maintenance*: instead of re-computing the entire XML view in response to source changes, only *changes to the* XML *view* are computed, which is often much smaller than the XML view and takes far less time to compute.

Schema-directed XML publishing and integration are, however, highly challenging. XML schemas are often complex, arbitrarily nested and even recursive, as commonly found in *e.g.,* biological ontologies [6]. This makes it hard to ensure that an XML view conforms to such a predefined schema. Add to this the difficulties introduced by XML constraints (*e.g.,* keys and foreign keys) which are often specified in a schema and have also to be satisfied by published and integrated XML data. These are further complicated by incremental XML view maintenance, which requires that changes to XML views should not violate the schema.

It is clear that new automated tools are needed. In particular, much in need is a uniform system to support XML publishing, integration and incremental view maintenance such that all these interact in the right way. However, no commercial systems are capable of supporting these. Indeed, while Microsoft SQL 2005 [9], Oracle XML DB [11] and IBM DB2 XML Extender [7] support XML publishing, they either ignore the schema-directed requirement (by using a fixed document template instead of an XML schema), or do not allow recursive XML schema. Worse still, none of these supports XML integration. When it comes to incremental XML view maintenance, to the best of our knowledge, no commercial DBMS provides the functionality. As for research prototypes, Clio [10] focuses on relational data integration based on schema mapping and does not address XML integration. While SilkRoute [5] and XPERANTO [12] are developed for XML publishing, they allow neither recursive XML schemas nor constraints. None of these systems supports XML integration or incremental view maintenance.

To this end we present PRATA, a system under development, that supports schema-directed XML publishing, integration and incremental view maintenance in a uniform framework. As depicted in Fig. 1, the system consists of three main components (modules): XML publishing, XML integration and incremental XML view maintenance.

To our knowledge, PRATA is the first and the only system that is capable of supporting all of these. It is worth

*Supported in part by EPSRC GR/S63205/01, GR/T27433/01 and BBSRC BB/D006473/1. Wenfei Fan is also affiliated to Bell Laboratories, Murray Hill, USA.

mentioning that while XML publishing is a special case of XML integration where there is a single source database, we treat it separately since it allows us to develop and leverage specific techniques and conduct the computation more efficiently than in the generic integration setting.

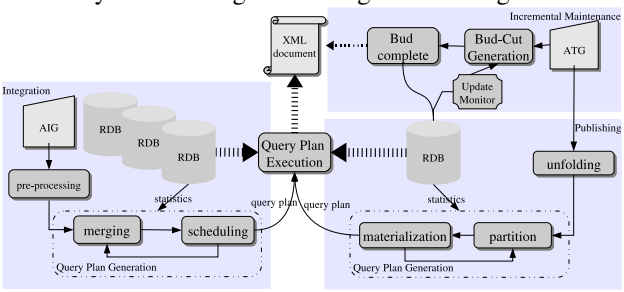


Figure 1. the System Architecture

2 XML Publishing

This module allows users to specify mappings from a relational database schema R to a predefined XML schema D, via a GUI and in a novel language *Attribute Translation Grammar* (ATG) that we proposed in [2]. Given an ATG σ and a database instance I of R, the system automatically generates an XML document (view) $\sigma(I)$ of I such that $\sigma(I)$ is guaranteed to conform to the given DTD D. This process is called *schema-directed publishing*.

Source relational schema R_0:

chapters(*chapter_id*, *name*)
receptors(*receptor_id*, *chapter_id*, *name*, *code*)
refs(*ref_id*, *chapter_id*, *year*, *title*)
cite(*ref_id*, *receptor_id*)

Figure 2. Relational schema R_0 for Receptors

We next demonstrate the ATG approach for publishing relational data in XML, by using a simplified example taken from the IUPHAR (International Union of Pharmacology) Receptor Database [8], which is a major on-line repository of characterisation data for receptors and drugs. We consider tables chapters, receptors, refs and cite shown in Fig. 2 (the primary keys for all tables are underlined). Table chapters stores the receptor families where each family has a *chapter_id* (primary key) and a *name*. Table receptors stores the *receptor_id*, *chapter_id*, *name* and *code* of receptors. Table refs stores publications on receptor families. Each reference has a *ref_id*, *title* and a *year* of the publication. It is associated with a unique receptor family through the attribute (foreign key) *chapter_id*. Finally, table cite records many-to-many relationships between receptors and references with *ref_id* and *receptor_id* (as foreign keys) pointing to references and receptors, respectively.

Target DTD D_0:

```
<!ELEMENT db          (family*)>
<!ELEMENT family*     (name, receptors, references )>
<!ELEMENT references (reference*)>
<!ELEMENT reference   (title, year)
<!ELEMENT receptors   (receptor*)>
<!ELEMENT receptor    (name, receptors)>
/* #PCDATA is omitted here.  */
```

Figure 3. DTD D_0 for publishing data of R_0

One wants to represent the relational Receptor data as an XML document that conforms to the DTD D_0 shown in Fig. 3. The document consists of a list of receptor families, identified by *chapter* ids. Each *family* collects the list of *receptor*s and the list of *ref*erences in the family (chapter). Note that *receptor* (thus D_0) is recursively defined: the *receptors* child of *receptor* element E_0 can also take an arbitrary number of *receptor* $E_1, E_2, \cdots, E_n$ as its children, which are the receptors related to E_0 through references. To avoid redundant information, if a *receptor* appears more than once in the reference hierarchy, the document stores it only once, *i.e.,* its first appearance.

The ATG σ_0 for publishing the Receptor data is given in Fig. 4, which extends the predefined DTD D_0 by associating semantic rules (*e.g.,* Q_1) to assign values to the variables (*e.g.,* $family). Given the Receptor Database, the ATG σ_0 is evaluated top-down: starting at the root element type of D_0, it evaluates the semantic rules associated with each element type encountered, and creates XML nodes following the DTD to construct the XML tree. The values of the variables $\$A$'s are used to control the construction. Now we give part of the ATG evaluation process.

(1) At each `receptors` element s, the target tree T is further expanded as follows. As $\$receptors.tag$ is assigned "0" from the family element's evaluation, the first branch of SQL query Q_2 is triggered. This branch finds the tuples of $receptor\_id$ and $name$ associated with each family f from the `receptors` relation, by using variable $\$receptors.id$ (passed from $\$family.chapter\_id$ in the last step) as a constant parameter. For each r of these tuples, a `receptor` child of s is created carrying r as the value of its variable.

(2) At each `receptor` element r, `name` element and `receptors` element s' are created (note that "1" is assigned to $\$receptors.tag$, and $\$receptor.receptor\_id$ is accumulated into $\$receptors.ids$ in this process).

(3) At each `receptors` element s', the second branch of SQL query Q_2 is triggered because $\$receptors.tag$ is now "1". This branch finds the tuples of $receptor\_id$ and $name$ from the `receptors` and `cite` relations related to receptor r that have *not* been processed in previous steps, by using variable $\$receptors.id$ and $\$receptors.ids$ as constant parameters. Note here that the variable $\$receptors.ids$ is used to decide whether or not a `receptor` has been processed earlier, and thus avoid redundant inclusion of the same `receptor` in the document T. For each r' of these tuples, a `receptor` child of s' is created carrying r' as the value of its variable, and the `receptor` element is in turn created as described in (2).

Details on the above conceptual level evaluation process and more effective techniques to generate efficient evaluation plans are show in [2]. These include query-partitioning and materializing intermediate results, in order to reduce communication cost between the publishing module and the underlying DBMS, based on estimates of the query cost and data size. We omit the details here due to the space con-

straint, but encourage the reader to consult [2].

Semantic Attributes: /*omitted*/
Semantic Rules:
db → **family***
Q_1: $family ← **select** chapter_id, name **from** chapters

family → **name**, **receptors**, **references**
$fname = ($family.name), $references = ($family.chapter_id),
$receptors = (0, $family.chapter_id, ∅)

receptors → **receptor***
Q_2: $receptor ← case $receptors.tag of
0: **select** receptor_id, name, $receptors.ids
from receptors
where chapter_id = $receptors.id
1: **select** a.receptor_id, a.name, $receptors.ids
from receptors a, cite b, cite c
where b.receptor_id = $receptors.id **and**
b.ref_id = c.ref_id **and**
b.receptor_id <> c.receptor_id **and**
a.receptor_id = c.receptor_id **and**
a.receptor_id **not in** $receptors.ids

receptor → **name**, **receptors**
$rname = ($receptor.name),
$receptors = (1, $receptor.receptor_id, $receptor.ids ∪ $receptor.receptor_id)

references → **reference***
Q_3: $reference ← **select** title, year
from refs
where chapter_id = $references.chapter_id

reference → **title**, **year**
$year = ($reference.year), $title = ($reference.title)

A → **S** /* A is one of *name, title, year* */
$S = ($A.val)

Figure 4. An example ATG σ_0

3 XML Integration

Extending the support for ATGs, this module provides a GUI for users to specify mappings from *multiple, distributed* databases to XML documents of a schema D, in our language *Attribute Integration Grammar* [1]. In addition, given an AIG and source databases, this module extracts data from the distributed sources and produces an XML document that is guaranteed to conform to the given DTD D and satisfy predefined XML contraints Σ.

As an example, suppose that the tables chapters and receptors in Fig. 2 are stored in a database DB1, while the tables refs and cite are stored in DB2. Now the IPUHAR department wants to generate a report about the receptors and corresponding publications. The report is required to conform to a fixed DTD D_1, which extends the D_0 of Fig. 3 as follows: elements `fid` and `rec_id` are added to the `family` production and `receptor` production as their first subelements respectively; and `ref_id` and `fid` are added to the `reference` production as its first two subelements (from table DB2 : refs we can see one `reference` can only relate itself to one `family`). In addition, for each `family`, the document is to collect all the `references` that are publications cited directly or indirectly by `receptors` in the `family`.

These introduce the following challenges. First, the integration requires multiple distributed data sources. As a result a single SQL query may access multiple data sources, referred to as *multi-source queries*. Second, receptors are defined recursively. Third, for each family, `references` can only be computed after the `receptors'` computation is completed since it is to collect all references in the `receptors` subtree, which is recursively defined and has an unbounded depth. In other words, this imposes a dependency between the `references` and the `receptors` subtrees. As a result the XML tree can not be simply computed as ATGs [1] by using a top-down method.

In addition, the XML report is also required to satisfy the flowing XML constraints:

ϕ_1: references(reference.ref_id → reference)
ϕ_2: db(reference.fid ⊆ family.fid)

Here ϕ_1 is a key constraint asserting that each subtree rooted at a `references` node, `ref_id` is a key of `reference` elements; and ϕ_2 is an inclusion constraint asserting that the families cited by references in the `db` must be presented in the `family` subtree of the `db`.

As remarked in Section 1, no commercial systems or research prototype can support schema-directed XML integration with respect to both DTD and XML constraints. The only effective technique for doing this is our *Attributes Integration Grammars* (AIGs) reported in [1], which is the underlying technique for the integration module of PRATA.

Based on multi-source query decomposition and constraints compilation PRATA is capable of integrating data from multiple sources and generating an XML document that is guaranteed to both conform to a given DTD and satisfy predefined XML constraints. PRATA also leverages a number of optimization techniques to generate efficient query evaluation plans [1]. In particular, it uses a cost-based scheduling algorithm, by taking into account dependency relations on subtrees, to maximize parallelism among underlying relational engines and to reduce response time. Due to the lack of space we omit the details here (see [1]).

4 Incremental XML Views Maintenance

This module maintains published XML viewsc $\sigma(I)$ based on our incremental computation techniques developed in [3]. In response to changes ΔI to the source database I, this module computes the XML changes ΔT to $\sigma(I)$ such that $\sigma(I \oplus \Delta I) = \Delta T \oplus \sigma(I)$,while minimizing unnecessary recomputation. The operator $\oplus$ denotes the application of these updates,

As remarked in Section 1, scientific databases keep being updated. Consider the XML view published by the ATG σ_0 of Fig. 4 from the the Receptor Database I specified by Fig. 2. Suppose that the relational Receptor database is updated by insertions $\Delta refs$ and $\Delta cite$ to the base relations $refs$ and $cite$ respectively. This entails that the new reference information must be augmented to the corresponding reference subtrees in XML views. Moreover, the insertions also increase the related receptors for some receptor nodes in the XML view and the augment may result in further expansion of the XML view. Given the recursive nature of σ_0,

this entails recursive computation and is obviously nontrivial.

The incremental algorithm in [3] is based on a notion of ΔATG. A ΔATG $\Delta\sigma$ is statically derived from an ATG σ by deducing and incrementalizing SQL queries for generating edges of XML views. The XML changes ΔT are computed by $\Delta\sigma$, and represented by of a pair of edge relations (E^+, E^-) , denoting the insertions(buds) and deletions(cuts). The whole process is divided into three phases: (1) a *bud-cut generation phase* that determines the impact of ΔI on *existing* parent-child(edge) relations in the old XML view T by evaluating a fixed number of incrementalized SQL queries; (2) a *bud completion phase* that iteratively computes newly inserted subtrees top-down by pushing SQL queries to the relational DBMS; and finally, (3) a *garbage collection phase* that removes the deleted subtrees.

The rational behind this is that the XML update ΔT is typically small and more efficient to compute than the entire new view $\sigma(I \oplus \Delta I)$. The key criterion for any incremental view maintenance algorithm is to precisely identify ΔT; in other words, it is to minimize recomputation that has been conducted when computing the old view T. Our incremental maintenance techniques make maximal use of XML sub-trees computed for the old view and thus minimize unnecessary recomputation. It should be pointed out that the techniques presented in this section are also applicable to XML views generated by AIG. We focused on ATG views just to simplify the discussion. Due to the limited space we omit the evaluation details here (see [3]).

5 PRATA: Features and Current Status

Taken together, PRATA has the following salient features, which are beyond what are offered by commercial tools or prototype systems developed thus far.

Schema conformance. PRATA is the first system that automatically guarantees that the published or integrated XML data conforms to predefined XML DTDs and schemas, even if the DTDs or schemas are complex and recursive.

Automatic validation of XML constraints. In a uniform framework for handling types specified by DTDs or schemas, PRATA also supports automatic checking of integrity constraints (keys, foreign keys) for XML.

Integration of multiple, distributed data sources. PRATA is capable of extracting data from multiple, distributed databases, and integrating the extracted data into a single XML document. A sophisticated scheduling algorithm allows PRATA to access the data sources efficiently in parallel.

Incremental updates. PRATA is the only system that supports incremental maintenance of *recursively defined* views, and is able to efficiently propagate changes from data sources to published/integrated XML data.

Novel evaluation and optimization techniques. Underlying PRATA are a variety of innovative techniques including algorithms and indexing structures for query merging [2], constraint compilation, multi-source query rewriting, query scheduling [1], and bud-cut incremental computation [3], which are not only important for XML publishing and integration, but are also useful in other applications such as multi-query evaluation and database view maintenance.

Friendly GUIs. PRATA offers several tools to facilitate users to specify publishing/integration mappings, browse XML views, analyze published or integrated XML data, and monitor changes to XML views, among other things.

The current implementation of PRATA fully supports (a) schema-directed XML publishing, and (b) incremental maintenance of XML views of a single source, based on the evaluation and optimization techniques discussed in previous sections. For XML schemas, it allows generic (possibly recursive and non-deterministic) DTDs, but has not yet implemented the support for XML constraints. A preliminary prototype of the system was demonstrated at a major database conference [4], and is deployed and evaluated at Lucent Technologies and European Bioinformatics Institute [4]. We are currently implementing (a) XML integration and (b) incremental maintenance of XML views of multiple sources. Pending the availability of resources, we expect to develop a full-fledged system in the near future.

References

[1] M. Benedikt, C. Y. Chan, W. Fan, J. Freire, and R. Rastogi. Capturing both types and constraints in data integration. In *SIGMOD*, 2003.

[2] M. Benedikt, C. Y. Chan, W. Fan, R. Rastogi, S. Zheng, and A. Zhou. DTD-directed publishing with attribute translation grammars. In *VLDB*, 2002.

[3] P. Bohannon, B. Choi, and W. Fan. Incremental evaluation of schema-directed XML publishing. In *SIGMOD*, 2004.

[4] B. Choi, W. Fan, X. Jia, and A. Kasprzyk. A uniform system for publishing and maintaining XML. In *VLDB*, 2004. Demo.

[5] M. F. Fernandez, A. Morishima, and D. Suciu. Efficient evaluation of XML middleware queries. In *SIGMOD*, 2001.

[6] GO Consortium. Gene Ontology. `http://www.geneontology.org/`.

[7] IBM. DB2 XML Extender. *http://www-306.ibm.com/software/data/db2/extenders/xmlext/*.

[8] IUPHAR. Receptor Database. `http://www.iuphar-db.org`.

[9] Microsoft. XML support in Microsoft SQL erver 2005, December 2005. *http://msdn.microsoft.com/library/en-us/dnsql90/html/sql2k5xml.asp/*.

[10] R. J. Miller, M. A. Hernández, L. M. Haas, L.-L. Yan, C. T. H. Ho, R. Fagin, and L. Popa. The Clio project: Managing heterogeneity. *SIGMOD Record*, 30(1):78–83, 2001.

[11] Oracle. Oracle Database 10g Release 2 XML DB Technical Whitepaper. *http://www.oracle.com/technology/tech/xml/xmldb/index.html*.

[12] J. Shanmugasundaram, E. J. Shekita, R. Barr, M. J. Carey, B. G. Lindsay, H. Pirahesh, and B. Reinwald. Efficiently publishing relational data as XML documents. *VLDB Journal*, 10(2-3):133–154, 2001.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Research Methods for Eliciting e-Research User Requirements

Florian Urmetzer, Mark Baker and Vassil Alexandrov

ACET Centre, The University of Reading

Abstract

e-Research is a relatively new and expanding means of undertaking multi-disciplinary on-line research. As such, the needs and requirements of the e-Research community are not yet fully understood and it is evident that it needs further investigation and study. This paper aims to provide basic guidance on the research methods that will be useful in order to gather information from the users of multi-disciplinary e-Research-based projects. The paper discusses quantitative and qualitative research methods with examples that have been drawn from social and computer science. We show that there are many ways of gaining information from users, but when used appropriately qualitative methods give more depth of information, whereas quantitative methods are scalable to the whole community.

1. Introduction

The need for interactive user surveys has been identified as the best means of accelerating the adoption of e-Research and ensuring the tools developed for the programme are more effective. This need is based on the opinion of consumers of e-Research resources, who wanted to alert others about matters from a user's point of view [1]. To incorporate user suggestions and requirements into projects a number mechanisms are needed for gathering information from the user base. Therefore, a discussion about the different methods available is essential. For example, Nielson [2] in experiments found that the difference in the choice of methods plays a large role in the quality of the findings. He evaluated a user interface, with three subjects. The test reported back five or six interface problems with well-suited methods used. However, the same evaluation would return only two or three usability problems with a poorly chosen methodology [3]. Similar results are expected for user enquiries.

Lederer and Prasad [4] report, in a study concerning the problems of cost estimations in software projects, that the four most cited reasons for overspending in projects were changes by users of the final software artefact; tasks that were left unobserved, users were not able to understand their own needs and there was a mismatch between users and programmers.

Part of the Sakai Virtual Research Environment (VRE) Demonstrator project [5] is investigating methods to gather information from the user community. This paper looks at research methods that can be used to gather focused user requirements information. Research work from the social science arena has been used to establish the basis of the methods employed. The computer science field was then investigated for examples of the adoption of the methods discussed. Finally, conclusions are made about the methods that appear to be most suitable for gathering focused information for the requirements of e-Research projects.

2. Research Methods

There are several methods possible for collecting information from a user base. They are generally split into two sub-groups. The first encompasses quantitative methods, such as user surveys. The second is based on qualitative methods, such as user interviews [6, 7].

2.1 Quantitative Methods

Quantitative methods are based on the collection of information in a numerical format that helps express the association between the real world and a theory or a thesis [6]. This can be split into two categories: first the experiment and second the survey [7]. Experiments are the defined treatment of variables and the measurement of the outcomes. This is seen as the grounding on which a cause and effect

relationship can be established, and, as a result, the outcome can be measured [8]. One example of this is to measure a new system's performance in relation to an existing system. This is known as a comparative experiment. Another option, which is named an absolute experiment, would be to test a system in isolation [9]. The measurement base here could be the time to accomplish an exercise that needs to be undertaken within a programme. For example, there are two groups, whose members can be argued to be similar to each other in certain characteristics. If one group learns to use a computer programme faster than the other, the experiment should allow the assumption that the programme used one group is easier to learn. In this example the variables are defined to be similar (group and task) and the measurement base (time to learn the programme) is the difference. The cause of this difference is the intuitiveness of the programme interface.

There are critical views towards the use of this theory. For example, Faulkner [9] quotes a study of 'morphic resonance', where the outcome of the experiments contradicted each other. Two researchers planned and conducted the experiment together, but started with opposing views of the outcome. The outcome of the experiment was then published in two contradicting papers. Faulkner [9] argued the reason for this was that the two individuals "hung on" to their opinions rather then to accept an interpretation that may destruct a view that one may be personally attached [9].

Surveys typically use questionnaires or structured interviews for the collection of data. They have a common aim to scale up and model a larger population [7]. Therefore, a survey can be seen as a set of questions creating a controlled way of querying a set of individuals to describe themselves, their interests and preferences [10].

Questionnaires are the most structured way of asking questions of participants. This is because the designer of the questionnaire decides upon the options for the answers. For example, a Yes or No; or tick one of the answers given. These multiple-choice answers make the analysis of the responses easy for the researcher [11]. Indeed, when it is seen in more detail this is not a very straight-forward approach, because as Kuniavsky [10] points out the questions have to be phrased in a correct way to avoid misunderstandings or double meaning. He uses the example of a financial Web page where the owners want to query how often their users purchase new stock. This may be well understood for most of their users. However, some people may talk about goods in a shop when using stock. Additionally, offering multiple-choice answers can be rather dull for the participants, because there are only pre-selectable answers [11]. This may be overcome by using multiple indicators, semantic differential scales or ranked order questions,

Which are often referred to as the Likert scale [7, 9].

There are also other tools like the semantic differential scale, in which the respondents get the task of ranking their opinion according to their importance. For example, using opposing terms like clumsy and skilful to indicate how these are perceived by the respondents [9].

Ranked order questions are another option for a questionnaire. The researcher here gives the option of ranking, for example, of importance of a programme on a PC. Options would include: a browser, word processor, image enhancer or messenger. The participant can then choose which option is the most important or the lesser important to the users.

Sample size

With surveys or structured interviews the sample size can be large. In many studies the sample size does not have a direct impact on the cost of the project, because the surveys can be sent via e-mail and the interviewer does not have to attend the filling in of the survey.

Bias

Bias is argued to be low when using quantitative methods. This is mainly reasoned with the absence of an interviewer, who is not able to influence the process. Also important is that the participant is guaranteed anonymity, which may prevent a bias in the response.

Quality of questioning:

The questions have to be short and precise. They have to be phrased very carefully so as not leaving room for miss-interpretation. As a result, to phrase the question in a certain way may have a negative impact on the answers.

Data quality:

The data quality is provided by the design of the questionnaire. When the questionnaire is designed, the answers are fixed and will be the ones counted. Therefore, there will be no room for further insight into causes or processes. Obviously, questionnaires can be poorly designed and may not gather useful information.

Negative Bias:

Bias may be introduced through selection of participants that make up the sample group. Also the phrasing of questions and the time and location may have an impact on the answers given, and perhaps introduce negative bias. For example, Jacobs *et. al.* [12] chose to use surveys

because their subjects where not accessible for direct interviews. Their study subjects where high level members of staff in an international organization.

2.2 Qualitative Methods

Qualitative methods contrast greatly with quantitative ones. Qualitative research is defined as a methodology that allows the interpretation of data more then counting predefined answers [6, 8, 13].
Quantitative methods may take longer and/or involve intensive contact with the situation of study. The situation of study can be real live situation, single or multiple people, organizations or populations [9]. The task of the researcher is to capture a situation in its full complexity and log the information found. The analysis then includes investigating the topics or subjects found and interpreting or concluding on their meaning from a personal as well as from a theoretical point of view [6].
It is however sometimes stated, that qualitative methods are not straight-forward and are often seen as more unclear or diffused as quantitative results [13].
The main methods for the collection of data are participant observation, qualitative interviewing, focus groups and language based methods [6, 8].
1. Participant observation is defined by an immersion of the researcher with the subject into the situation to be studied [8]. This will allow the collection of data in an unobtrusive way by watching the behaviour and activities of the subject. Observation is often used in user interaction studies, looking for example at Human Computer Interfaces [13].
2. Qualitative interviewing can be defined as a dialogue, where one of the participants (the researcher) is looking for answers from a subject [14]. Qualitative interviewing is often split into unstructured and semi-structured interviewing. In the unstructured case the interviewer is only equipped with a number of topics that should, but do not have to be, discussed in the interview. Whereas in the semi-structured case the interviewer has a series of questions, which should be addressed, but the sequence may be varied. Most importantly, compared to quantitative interviewing, both interview types allow additional questions on the basis of the response to a question [8].
3. Focus groups consist of a small number of people, chosen to participate in an informal discussion guided by a moderator. The group's task is to discuss defined subjects, opinions and knowledge [15-17]. The advantage of this kind of interaction is that the opinion of one individual can cause further dialogue from others. Moreover, a larger number of opinions, attitudes, feelings or perceptions towards a topic are voiced by the participants, this provides a broader overview of the discussed topic [11, 13].
4.Language-based methods, such as content and conversation analysis; work on the basis of a newspaper article or transcribed conversation. Their main aim is to uncover lower level structures in conversations or texts [8]. This, however, is not considered here, mainly because there were no examples found of this method.

Qualitative methods bring forward positive and negative aspects to the research conducted using them. Accounts of the positive aspects are the depth and the quality of information [6, 15, 18, 19] gathered. As Punch puts it, "qualitative interviews are the best way to understand other people". Therefore, perceptions, meanings, situations descriptions and the construction of realities can be best recorded in a qualitative setting [9]. A negative impact is the dependency on the moderator, who is seen as important in the process of asking the questions and interpreting the data [6, 16]. Holzinger [20], for example, used forty-nine participants for a study looking at the creation of an interface using mainly qualitative methods to gain information from the users. The researchers described methods as useful for identifying the users' needs. Rose et. al. [17] reported on using focused interviews and focus groups to investigate the use of a medical decision support tool. The focused interviews where used to identify tasks and their goals and the focus groups to investigate in which situations the tool would be used. Konio [21] concludes on the comparison of three software projects, which have been based on using focus groups, that the outcome was positive. However, they point out that a focus group needs to be run by a trained person. Similar findings where reported by Wilson [22]. His team has been conducting focus groups to gain information about a VRE [22].

3. Conclusion

As outlined in this paper, there are a number of approaches for gaining information from the e-Research community. In the case of gathering information about requirements towards a VRE, aspects of bias, sampling and quality of the

information gathered have to be taken into account.

Furthermore, quantitative findings are not as dependent on the moderator as quantitative methods. However, the depth of information given by the quantitative methods is not as elaborate as qualitative methods, because of its predefined questions and answers. Moreover, when using quantitative methods there is normally not the option to elaborate on questions. Therefore, the developers would have to have thought about all the requirements needed by the user group beforehand, which can be strongly argued to be unlikely case. Therefore, quantitative methods can restrict the enquiring requirements of the e-Research community.

When looking at qualitative research methods, there are opportunities to ask groups of users or individuals. This provides the opportunity to understand and query in detail, what functionality they want from their project. Therefore the outcomes will give more information about the background and reasoning of user requirements.

This depth and quality of information would not be achievable using quantitative methods due to the fixed questioning structure. When examining the difference between individual qualitative interviews and group inquiries, the discussion evolving from the group will enable to greater detail to be recorded and individuals can clarify requirements by discussing them with peers rather then with an IT specialist [16, 18]. This may prevent the changing of ideas over time. Through the multiplication of focus groups, the importance of functionality can be verified and therefore arguments can be cross-referenced as described by [8].

The arguments above qualitative methods seem to be the best path to pursue for the user needs gathering in the VRE programme. However, it is acknowledged that if the enquiry is done with a low level of motivation and attention to detail, the information gathered may be bias and therefore the tasks will not be well informed through the users.

3. References

1. Marmier, A. *The eMinerals minigrid and the National Grid Service: a user's perspective*. in *The All Hands Meeting*. 2005. Nottingham, UK.
2. Nielsen, J., *Usability engineering*. 1993, Boston; London: Academic Press. xiv, 358.
3. Mack, R.L. and J. Nielsen, *Usability inspection methods*. 1994, New York; Chichester: Wiley. xxiv, 413.
4. Lederer, A.L. and J. Prasad, *Nine Management guidelines for better cost estimating.* Communications of the ACM, 1992. **35**(2): p. 51-59.
5. University, L. *VRE Demonstrator*. [Internet] 2006 [cited 2006 01 May 06]; http://e-science.lancs.ac.uk/vre-demonstrator/index.html].
6. Bryman, A., *Social research methods*. 2nd ed. 2004, Oxford: Oxford University Press. xiv, 592.
7. Creswell, J.W., *Research design: qualitative, quantitative, and mixed method approaches*. 2nd ed ed. 2003, Thousand Oaks, Calif.; London: Sage. xxvi, 246.
8. Punch, K., *Introduction to social research: quantitative and qualitative approaches*. 2nd ed. 2005, London: SAGE. xvi, 320.
9. Faulkner, X., *Usability engineering*. 2000, Basingstoke: Macmillan. xii, 244.
10. Kuniavsky, M., *Observing the user experience: a practitioner's guide to user research*. Morgan Kaufmann series in interactive technologies. 2003, San Francisco, Calif.; London: Morgan Kaufmann. xvi, 560.
11. Gillham, B., *Developing a questionnaire*. Real world research. 2000, London: Continuum. ix, 93.
12. Jakobs, K., R. Procter, and R. Williams. *A study of user participation in standards setting*. in *Conference companion on Human factors in computing systems: common ground*. 1996. Vancouver, British Columbia, Canada: ACM Press.
13. Seaman, C.B., *Qualitative methods in empirical studies of software engineering.* Software Engineering, IEEE Transactions on, 1999. **25**(4): p. 557.
14. Gillham, B., *The research interview*. Real world research. 2000, London: Continuum. viii, 96.
15. Litosseliti, L., *Using focus groups in research*. Continuum research methods series. 2003, London: Continuum. vii, 104.
16. Nielsen, J., *The use and misuse of focus groups.* Software, IEEE, 1997. **14**(1): p. 94.
17. Rose, A.F., et al., *Using qualitative studies to improve the usability of an EMR.* Journal of Biomedical Informatics, 2005. **38**(1): p. 51-60.
18. Morgan, D.L., *Focus groups as qualitative research*. 2nd ed. 1997, Thousand Oaks; London: Sage. viii, 80.

19. Wilkinson, D. and P. Birmingham, *Using research instruments: a guide for researchers*. 2003, New York; London: Routledge/Falmer. xiv, 175.
20. Holzinger, A., *Rapid prototyping for a virtual medical campus interface.* Software, IEEE, 2004. **21**(1): p. 92.
21. Kontio, J., L. Lehtola, and J. Bragge. *Using the focus group method in software engineering: obtaining practitioner and user experiences*. 2004.
22. Urmetzer, F., *Qualitative enquiry: the use of focus groups in The University of Nottingham VRE Implementation Evaluation*. 2006, unpublished: Nottingham.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

A generic approach to High Performance Visualization enabled Augmented Reality

Chris J. Hughes[1a] Nigel W. John[a] Mark Riding[b]

[a]University of Wales, Bangor
[b]Manchester Computing, University of Manchester, UK

Abstract

Traditionally registration and tracking within Augmented Reality (AR) applications have been built around limited bold markers, which allow for their orientation to be estimated in real-time. All attempts to implement AR without specific markers have increased the computational requirements and some information about the environment is still needed. In this paper we describe a method that not only provides a generic platform for AR but also seamlessly deploys High Performance Computing (HPC) resources to deal with the additional computational load, as part of the distributed High Performance Visualization (HPV) pipeline used to render the virtual artifacts. Repeatable feature points are extracted from known views of a real object and then we match the best stored view to the users viewpoint using the matched feature points to estimate the objects pose. We also show how our AR framework can then be used in the real world by presenting a markerless AR interface for Transcranial Magnetic Stimulation (TMS).

Keywords: Augmented Reality (AR), High Performance Visualization (HPV), Grid

1. Introduction

Augmented Reality (AR) applications superimpose computer-generated artefacts into the user's view of the real world. These artefacts must be correctly orientated with the viewing direction of the user who typically wears a suitable Head Mounted Display (HMD). AR is a technology growing in popularity in medicine, manufacturing, architectural visualization, remote human collaboration, and the military [1, 2].

To create the illusion of a virtual artefact within the real world, it is essential that the virtual object is accurately aligned and that the computer graphics are presented in real time. Most of the existing solutions involve the use of bold markers that contain contrasting blocks of colour and shapes making them easily identifiable using computer vision techniques. To align virtual artefacts into the real world three main stages are required – see figure 1. Firstly we need to examine the user's viewpoint and identify where our virtual objects belong in the scene. Secondly we need to track the object to ensure that we have aligned the object to the correct position. Finally we use pose estimation to calculate the orientation of the object so that we can align it with the real world.

The Human Interface Technology Laboratory at the University of Washington has developed the ARToolkit, a software library providing the tools for creating marker based AR applications. The ARToolkit has provided the foundation for many of the early developments in AR and make it possible to rapidly produce AR applications using an inexpensive webcam and an average specification PC [3].

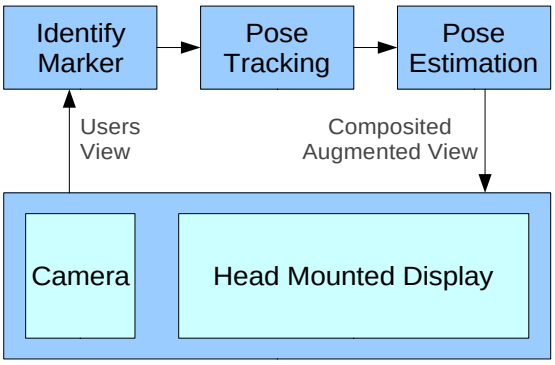


Figure 1: The three stages involved in AR

Moehring, Lessig and Bimber show that when using markers that have high contrast (for example, they use a bold type on a white background), little processing power is actually required to estimate the pose of an object even with the poor capture quality and low processing capability of a standard consumer mobile phone. [4]

Although the use of markers in optical tracking enables the pose estimation to be calculated relatively easily, having to use specific markers can limit the possible AR tools that can be made available. Therefore there are now many examples of AR solutions which do not require markers [5, 6, 7].

[1] Contact Author: chughes@informatics.bangor.ac.uk

In order to be successful, marker-less tracking is not only more computationally intensive, but also requires more information about the environment and the structure of any planes or real objects that are to be tracked. In this paper we present a generic solution that does not rely on the use of markers, but rather feature points that are intelligently extracted from the users view. We also provide a solution to the computationally intensive task of pose estimation and the rendering of complex artefacts by exploiting remote HPV resources through an advanced environment for enabling visual supercomputing – the e-Viz Project [8].

2. Robust feature point detection

In order to align our virtual object with the real world, we first need to define the object in the users view. During a calibration stage the user is given the opportunity to specify where the object exists within the viewpoint. We use a robust feature point detection algorithm to identify the points that can be repeatedly identified within the space occupied by the virtual object and use this information to estimate the objects position and orientation.

There are many existing methods for extracting feature points, most of which are based on corner detection algorithms. Since the 1970's many feature point detectors have been proposed and there is still work today to improve their accuracy and efficiency. There are three main methods for detecting feature points in images, which stem from the following methods: edge-detection, topology and autocorrelation [9, 10].

It is generally accepted that the autocorrelation methods yield the most repeatable results and they all follow the following three steps:

1. For each point in the input image a Cornerness value is calculated by the operator, and relates to how likely it is believed that that point is a corner.

2. A threshold value is used to disregard any points that are identified but are not strong enough to be true corners. The Cornerness value of these points is then typically set to zero.

3. Non-maximal suppression sets the Cornerness value for each point to zero if its cornerness value is not larger than the cornerness measure of all points within a certain distance. This ensures that we only find maximum points and so we can then assume that all non-zero points are corners.

2.1 Moravec/ Harris algorithms

A very basic algorithm was proposed by Moravec in 1977 as part of his work on machine vision enabled robotics [11, 12]. He proposed that a point could be identified as a feature point if there was a significant intensity variation in each direction from that point. Although this algorithm provides basic feature detection without being too computationally intensive, it is not repeatable as the points it finds are only repeatable when the edges are at 45° or 90° to the point being evaluated. The Harris algorithm [13] improves the Moravec algorithm but at a significant cost to the computational requirements. It becomes more robust by changing the way intensity variation is calculated between each pixel and its neighbours by allowing for edges that are not at 45° or 90° to the point being evaluated.

The Harris algorithm uses first order derivatives to measure the local autocorrelation of each point. A threshold value is then used to set all of the weaker points to zero leaving all of the non zero points to be interpreted as feature points- see figure 2.

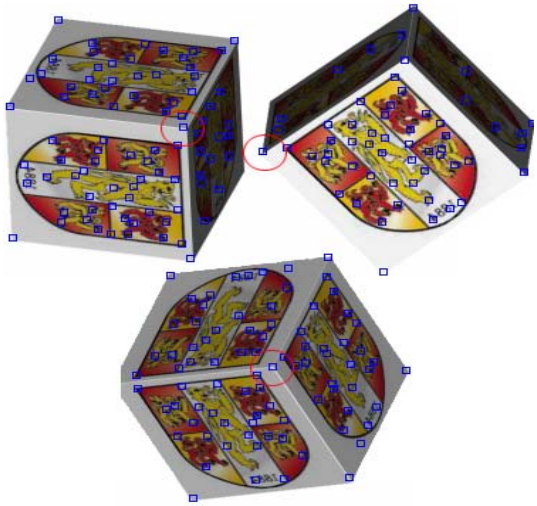


Figure 2: The Harris algorithm detecting repeatable feature points. The circles show a single point which has been accurately repeated in each movement of the cube.

3. Pose Tracking and Estimation

Our application uses the calibration information to train a Haar Classifier [15]. We have extended previous work with Haar Classifiers [16] by using multiple calibration views allowing the detector to not only be more robust but also to continue to track different sides of an object. We also maintain real-time performance even with the increased computational load by distributing the pose estimation as part of our visualization pipeline.

4. Utilizing HPV with e-Viz

The e-Viz project [8] is currently under development and aims to provide a generic flexible infrastructure for remote visualization using the Grid [17]. e-Viz address many of the issues involved in HPV [18] using a combination of intelligent scheduling for new rendering pipelines and the monitoring and optimisation of running pipelines, all based on information held in a knowledge base. This provides an adaptive visualization service that provides rendered graphics reliably without the application or user even being aware of what resources are being used. It also allows the application to render graphics in real time at a resolution that would normally be too high for the client machine.

We have followed two paths for implementing our application with e-Viz:

- **Rendering the graphics with e-Viz**

The first implementation simply uses e-Viz to render the virtual artefacts present in our AR view. The user's viewpoint is captured by the local machine and the pose estimation is calculated locally. The pose transformation is used to steer the e-Viz visualization pipeline, which in the background sends the transformation information to an available visualization server. Our client then receives the rendered image and composites it locally into the users view.

- **Distributing the pose estimation module as part of the visualization pipeline.**

In order to fully take advantage of the e-Viz resources the second implementation moves the pose estimation module onto the e-Viz visualization pipeline. In this case the e-Viz visualization is steered directly by the video stream of the users view. e-Viz distributes the pose estimation module to a suitable and available resource. The pose estimation module then steers the visualization pipeline and returns the final view back to the user after compositing the artificial rendering into the real scene.

4.1 The e-Viz API

e-Viz provides a client application that can be used to control a remote visualization pipeline as well as providing a viewer for the remotely rendered graphics to be returned to the user. It also provides an API that allows users to develop their own client applications which can utilize the e-Viz resources.

The e-Viz framework uses a web service to decide which hardware and software to make available to the client, based upon what resources are needed and what resources are available. The broker uses a knowledge base to store the status of the available servers and inventory what resources they are capable of providing. The client can interact with the Broker by the use of gSOAP calls, which will tell the Client which visualization servers to connect to.

There are generally multiple visualization servers within the e-Viz environment. Having discovered which visualization servers to use, the Client application uses a Grid middleware (such as GT2) to connect to the remote server and run the visualization task. By providing a wrapper to different visualization applications it makes it possible to execute your visualization pipeline on any visualization server regardless of what visualization software it is running.

5. Exemplar application

Transcranial Magnetic Stimulation (TMS) is the process in which electrical activity in the brain is influenced by a pulsed magnetic field. Common practice is to align an electromagnetic coil with points of interest identified on the surface of the brain, which can be stimulated helping researchers identify further information about the function of the brain. TMS has also proved to be very useful in therapy and had positive results with treating severe depression and other drug resistant mental illnesses such as epilepsy [19, 20].

In previous work we developed an AR interface for TMS using an optical tracking system to calculate the pose of the subjects head relative to user's viewpoint [21]. We are now developing a new AR interface that uses our generic framework – see figure 3. By removing the need for expensive optical tracking equipment our software will provide an inexpensive solution, making the procedure more accessible to training and further research.

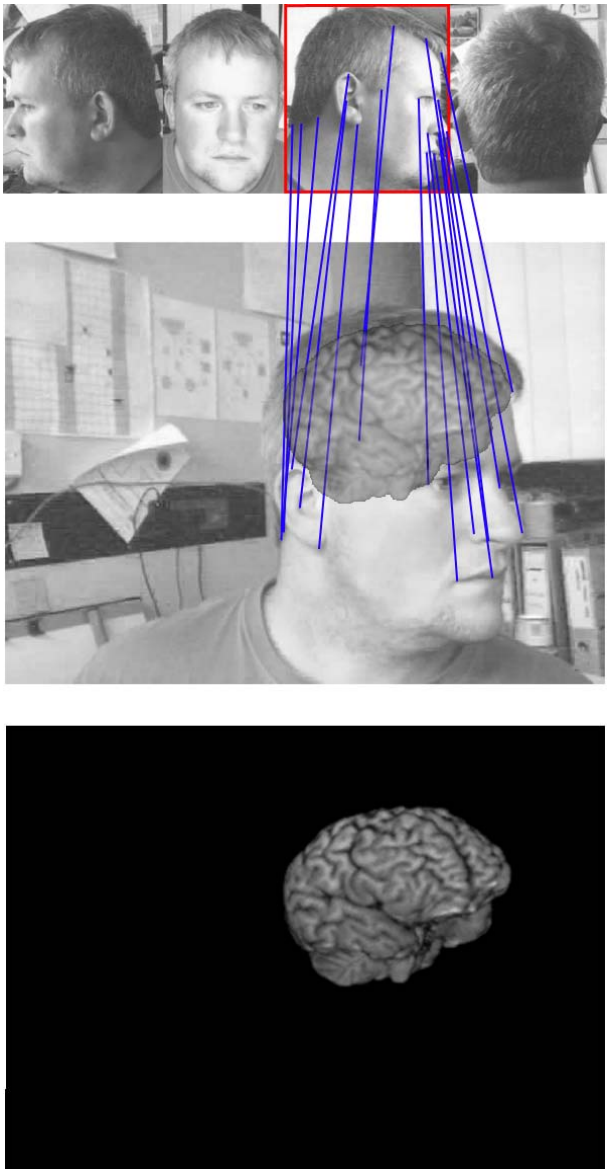


Figure 3: (a) The four images used to train the Haar classifier with the best match highlighted. (b) The real-time users view with lines illustrating some of the matched feature points. (c) The e-Viz remote rendering view.

Our research has shown that although although an average desktop PC does struggle with the pose estimation, using remote resources can ensure real-time performance. Provided the visualization server is appropriate for the rendering task the e-Viz framework is able to return the rendered artefact to the user at a

reliable 15 FPS, where there is little latency. On congested networks e-Viz uses stricter compression algorithms at a cost to the image quality to try and maintain these usable frame rates.

6. Conclusions

In conclusion we have found that our approach to producing a framework for AR has been very successful, provided that optimum conditions are available. Problems occur when trying to run the pose estimation locally. It is simply too computationally intensive and so can not keep up with the real time video. Distributing this calculation to a more powerful grid resource has solved this problem.

Future work will concentrate on improving the efficiency and reliability of the feature point detection algorithms, ensuring that we have more accurate pose estimation between frames. We also need to introduce heuristics that will help predict the position of the virtual artefact, even if we are unable to calculate the pose of the object, by building up a knowledge base of previous frames.

Acknowledgements

This research is supported by the EPSRC within the scope of the project: “An Advanced Environment for Enabling Visual Supercomputing” (GR/S46567/01). Many thanks to the e-Viz team at Manchester, Leeds and Swansea for their help and support.

We would like to thank Jason Lauder and Bob Rafal from the School of Psychology, University of Wales, Bangor, for allowing us access to their TMS laboratory and for supplying data from past experiments.

References

[1] Azuma, R. T., “A survey of Augmented Reality, Presence: Teleoperators and Virtual Environments”, Vol 6, No. 4 (August), pp. 355-385, 1997

[2] Azuma, R. T., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B., "Recent Advances in Augmented Reality", IEEE Computer Graphics and Applications 21, 6 (Nov/Dec 2001), 34-47.

[3] Kato, H., Billinghurst, M., “Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conference System”, In 2nd IEEE and ACM International Workshop on Augmented Reality , San Francisco USA, 1999 pp.85-94.

[4] Mohring, M., Lessig, C., Bimber, O., “Video See-Through AR on Consumer Cell-Phones,” ismar, pp. 252-253, Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04), 2004.

[5] Comport, A.I., Marchand, É., Chaumette, F., “A real – time tracker for marker-less
augmented reality”, In proceedings of IEEE and ACM International Symposium on Mixed
and Augmented Reality, pp. 36-45, 2003.

[6] Genc, Y., Riedel, S., Souvannacong, F., Navab, N. “Marker-less tracking for AR: A learning-based approach”, In proceedings of IEEE and ACM International Symposium on
Mixed and Augmented Reality, pp. 295-304, 2002.

[7] Striker, D., Kettenbach, T., “Real-time and marker-less vision-based tracking for
outdoor augmented reality applications”, In proceedings of IEEE and ACM International
Symposium on Augmented Reality, pp. 189-190, 2001.

[8] Riding, M., Wood, J., Brodlie, K., Brooke, J., Chen, M., Chisnall, D., Hughes, C., John, N.W., Jones, M.W., Roard, N., "e-Viz: Towards an Integrated Framework for High Performance Visualization", Proceedings of the UK e-Science All Hands Meeting 2005, EPSRC, ISBN 1-904425-53-4, pp1026-1032

[9] Zitova, B., Flusser, J., Kautsky, J., Peters, G., “Feature point detection in mulitframe images”, Czech Pattern Recognition Workshop, February 2-4, 2000.

[10] Hodges, K. I., “Feature-Point Detection Using Distance Transforms: Application to Tracking Tropical Convective Complexes”, American Meteorological Society, March 1998.

[11] Moravec H.P, “Towards Automatic Visual Obstacle Avoidance”, Proc. 5th International Joint Conference on Artificial Intelligence, pp. 584, 1977.

[12] Moravec H.P., “Visual Mapping by a Robot Rover”, International Joint Conference on Artificial Intelligence, pp. 598-600, 1979.

[13] Harris, C., Stephens, M., “A Combined Corner and Edge Detector”, Proc. Alvey Vision Conf., Univ. Manchester, pp. 147-151, 1988.

[14] Trajkovic, M., Hedley, M., “Fast Corner Detection”, Image and Vision Computing, Vol. 16(2), pp. 75-87, 1998.

[15] Wilson, P. I. and Fernandez, J. 2006. Facial feature detection using Haar classifiers. J. Comput. Small Coll. 21, 4 (Apr. 2006), 127-133

[16] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features.
In Conference on Computer Vision and Pattern Recognition, pages 511–518, 2001.

[17] Foster, I. and Kesselman, C. (eds.). The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999.

[18] Riding, M., Wood, J., Brodlie, K., Brooke, J., Chen, M., Chisnall, D., Hughes, C., John, N.W., Jones, M.W., Roard, N., “Visual Supercomputing - Technologies, Applications and Challenges”, Computer Graphics Forum, Vol. 24 Issue 2, 2005 pp217-245

[19] Ettinger, G. et al., “Experimentation with a Transcranial Magnetic Stimulation System for Functional Brain Mapping”, Medical Image Analysis, 2(2):133 - 142,1998

[20] Walsh V, Rushworth M., “A primer of magnetic stimulation as a tool for neurophysiology”, Neuropsychologia, Peergamon, 1999 Feb;37(2):125-35.

[21] Hughes, C. J., John, N.W., "A flexible infrastructure for delivering Augmented Reality enabled Transcranial Magnetic Stimulation", In Proc. Medicine Meets Virtual Reality 14, January 24-27, 2006, Long Beach, California, pp219-114

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Modelling Rail Passenger Movements through e-Science Methods

Jeremy Cohen[a], Claire James[b], Shamim Rahman[b], Vasa Curcin[a], Brian Ball[b], Yike Guo[a], John Darlington[a]

[a]London e-Science Centre, Imperial College London, South Kensington Campus, London SW7 2AZ, UK
[b]AEA Technology Rail, Central House, Upper Woburn Place, London WC1H 0JN, UK
Email: jeremy.cohen@imperial.ac.uk

Abstract. The UK's railway network is extensive and utilised by many millions of passengers each day. Passenger and train movements around the network create large amounts of data; details of tickets sold, passengers entering and exiting stations, train movements etc. Knowing how passengers want to use the network is critical in planning services that meet their requirements. However, understanding and managing the vast amounts of data generated daily is not easy using traditional methods. We show how, utilising e-Science methods, it is possible to make understanding this data easier and help the various stakeholders within the rail industry to more accurately plan their operations and offer more efficient services that better meet the requirements of passengers.

1 Introduction

The UK's railway network is large and complex. To plan and set strategy, government and other stakeholders need to know how passengers travel on the rail network, and the level of demand for rail services now and in the future. To manage such a network is a difficult task with operators needing to know how frequently to run trains, to which locations and at what times of day. If there is a demand for more frequent services or longer trains on certain routes, it is in an operator's interests to provide these. However, contrary to the belief of the average traveller, simply adding a new service or running a longer train is not simple. The timetables that control the movements of trains around the network require complex computational models to produce. But how do operators know when these enhancements or changes are, or will be, necessary? We aim to tackle this question through our work in the Department for Transport (DfT) funded project, under the Horizons research programme, "Effective Tracking of Rail Passenger Journeys".

We begin by looking at the types of rail network data that are available and how these are useful to the project. We then introduce the Discovery Net [8] platform that we have adopted to aid our research and show how it has been used for the necessary data mining and modelling. We conclude by presenting the results of the work.

2 Rail Data

There are currently many types of passenger journey data collected in the rail industry. The data may take a snapshot view of passenger journeys, where the data is collected manually (surveys and counts on trains), or be a continuous feed if collected electronically (through automatic gates or ticket sales). Each source of data is characterised by its attributes such as the method of capture, collation and processing, the size of the dataset, and its coverage (in terms of time period and geography). The challenge lies in effectively combining these data sources in a way to derive the most benefit in terms of understanding passenger movements.

2.1 Types of rail data

Some of the types of data available are:

- **Guards/Conductor Counts:** Manual counts taken by train guards/conductors.
- **Terminus Counts:** Manual counts carried out at terminus stations.
- **Automatic Ticket Gates:** Data collected by automatic ticket gates installed at many stations.
- **Automatic Passenger Counts:** Provided by weighing equipment fitted on more modern trains.
- **Ticket Sales:** Centrally collected data on National Rail tickets sold at stations and some third party ticket sales agencies.

- **London Area Travel Survey (LATS):** Effectively a census on travel patterns within the London area. Carried out once every ten years, most recent survey conducted in 2001.

2.2 Use Cases

Through consultation with key industry stakeholders, a number of use cases were formulated to cover the various ways in which passenger journey data is utilised by different users within the rail industry. These use cases helped highlight key areas where enhanced data was felt to be of greatest importance, as well as focusing the research into those areas where combining the existing passenger journeys data would enable types of analysis and decision-making currently limited by the nature of the separate data sources.

One example of a use case is the Route Utilisation Studies (RUS) carried out by Network Rail to develop efficient capacity plans that match stakeholders' requirements. These studies demand a detailed understanding of travel patterns on particular train routes, including information about train loads on these routes. This is a prime example of a recurring process where multiple sources of data are used to reach a decision. By providing easily accessible, accurate data, a centralised journey system would increase efficiency further through access to better quality information, and reduce the costs of carrying out these studies.

3 Discovery Net

Discovery Net [1, 6, 2, 8] is an EPSRC e-Science Pilot Project based at Imperial College London. The project has developed a service-based computing infrastructure for high throughput informatics that supports the integration and analysis of data collected from various high throughput devices. This infrastructure has been designed and implemented based on a workflow model, allowing the composition of data analysis services and resources declared as Web/grid services. The Discovery Net infrastructure is currently used by research scientists worldwide to conduct complex scientific data analysis in three important research areas: Life Sciences, Geohazard Modelling, and Environmental Modelling.

The problem posed by the Horizons research project is categorised by a set of key properties that Discovery Net is particularly well suited for working with – Large quantities of data, geographically distributed data sources, a wide variety of different questions that need to be answered, visualisation of results and dissemination to both technical and operational staff.

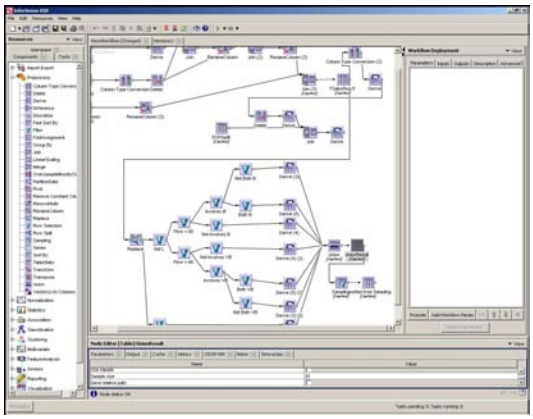


Fig. 1. The InforSense KDE client interface displaying one of the workflows developed for this project.

Given the relatively short 1-year duration of the project, development of a bespoke platform would not be possible. The Discovery Net platform provides all the required features and was therefore adopted as the underlying framework for our system. We are utilising the InforSense KDE software which is based on the research outputs of the Discovery Net project.

4 Extracting information through Discovery Net

In order to effectively manage railway network capacity, it is necessary to know how passengers use the network. There are many different ticket types, routes and operators. In the case of smaller stations or less complex routes it may be possible to know the operator of the train that a passenger takes, and perhaps even the specific train but in the case of large cities such as London or Birmingham, knowing which train a passenger has taken from a major interchange station may be almost impossible. It has been shown in section 2 that there is no shortage of data available on many routes, the challenge is to be able to link and interpret the data to reliably understand how passengers use the services available to them.

Discovery Net enables us to rapidly prototype possible solutions and extract the more detailed information we require, from the available data. This work utilises many of the ideas behind Computational Grids [4, 5] that underpin

e-Science and promote easier access to high performance computing resources for carrying out large-scale science.

The domain specific knowledge of the required analysis is held by the project members from AEA Technology Rail. Our aim is to encapsulate this knowledge within the computational platform, in the form of workflows. We worked in pairs consisting of an e-Science and rail data specialist in order to transfer the analysis description into Discovery Net workflows. The collaborative nature of the project combined with time constraints led us to see this as the most practical solution.

A workflow is a description of the processes and data/control flows required to carry out a task. The building blocks of workflows are components. In Discovery Net, a component can be a data source or a process applied to some data. A large toolbox of processes is available within the system and custom components can be built. One way of developing a custom component is to use the scripting language Groovy [7]. Workflows are built in the InforSense KDE client software using an intuitive drag-and-drop interface to select components from the toolbox and add them into the workflow.

4.1 System Platform

A computational platform has been set up specifically for the project. The InforSense KDE software system has been deployed on a 48 processor Sun Sparc IV based server operated by the London e-Science Centre. This multiprocessor system allows for faster operation of the InforSense engine when executing workflows that exhibit parallel features. The InforSense software can connect to distributed data sources running on different servers in different locations. In our demonstration environment, data is split between the London e-Science Centre's Oracle 10g server and data that has been imported directly into the InforSense server.

4.2 Workflow execution

Figure 1 shows the InforSense KDE client interface displaying one of the workflows we have developed. Execution of the workflows can be carried out on a multiprocessor system which offers significant speed benefits over a standard system. The Discovery Net architecture is designed to take advantage of parallel machines by determining elements of a workflow that can be executed in parallel and then taking advantage of multiple processors on a system by executing these components concurrently. To simulate a fully distributed e-Science architecture, we have used an external Oracle database containing warehoused data as one of the data sources within our system.

5 Results

This work has shown how rail datasets can be combined together in a systematic way, using the latest e-Science technologies, in order to maximise the useful information derived, and that this combination can provide a vital contribution to tackling a range of standard rail industry questions / studies with increased speed and accuracy. The project outputs can be separated into the following elements:

1. Comprehensive and consistent documentation of all major sources of passenger journey data existing in the rail industry.
2. Demonstration of how these may be combined to maximise the useful information, for two particular examples:
   (a) The ticket sales database (LENNON) and the London Area Travel Survey.
   (b) The national rail timetable and on-train automatic passenger counts.
3. Confirmation that a wide range of rail industry questions/studies can be improved by better combination of passenger data sources.
4. Incorporation of an example combination of rail data sources into a system, in order to show how the wide range of different data sources may be combined systematically.
5. Illustrative prototype of the user interface, in the InforSense KDE software.

Presentation of the results of our workflow processing utilises InforSense KDE's Web-based portal into which workflows can be published. A user logs into the system and is able to see the set of workflows in their Web browser to which they have been granted access privileges. The user can then change some attributes and execute the workflows available to them but may not modify these workflows.

Completed workflows, based on our rail industry use cases, provide a table of results and one or more visualisation options to allow industry decision makers to gain a simple visual view of the results of the analysis task.

5.1 Accuracy

When linking data sources in order to interpolate and extract new information, accuracy of the original data sources becomes very significant. By adding data sets together, the inaccuracies of each data set are multiplied together, resulting in data that may have such a high level of inaccuracy that it is effectively useless. In order to combat this risk, a significant amount of time was devoted to carrying out a comprehensive assessment of the accuracy of data sources and linked data that was to be used. These assements were carried out with reference to the quality dimensions used by the Office for National Statistics (ONS) and defined for the European Statistical System [3].

We believe that although inaccuracies in source data are a significant problem in this kind of work, our proposed solution takes these risks into account and makes an attempt to quantify them. By working to eliminate accuracy issues in key areas that we have identified, the entities capturing the data we have used can provide more accurate results from our framework in the future.

6 Conclusion

We have aimed to show in a very concrete manner how e-Science research can be applied in the real world to tackle an important problem. Setting the strategy for, and the management of, the UK's rail infrastructure is a difficult task, partly because each of the situations that need to be considered is so large. Many of these problems are therefore well suited to e-Science solutions.

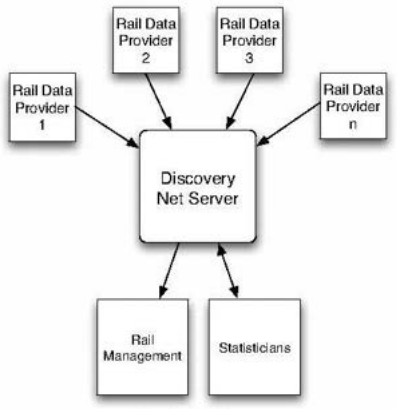


Fig. 2. Data from distributed sources is processed within the Discovery Net system and disseminated to the relevant industry staff.

Using the Discovery Net platform, we have built workflows to represent a set of key analysis tasks that help to answer questions rail industry managers need to address. Our flexible solution provides two points of entry to satisfy the requirements of both technical and operational staff and can present answers to important questions faster than existing solutions, or in some cases, that cannot currently be answered by existing systems.

Acknowledgements: We would like to thank Ian Hawthorne and the Department for Transport who have funded this project work under the Horizons Research Programme. We would also like to thank the Train Operating Companies and other organisations who have provided data samples without which this project could not have taken place.

References

1. V. Curcin, M. Ghanem, Y. Guo, M. Kohler, A. Rowe, J. Syed, and P. Wendel. Discovery net: Towards a grid of knowledge discovery. In *KDD-2002: The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 2002. `http://www.discovery-on-the.net/documents/kdd-DNET.pdf`.
2. V. Curcin, M. Ghanem, Y. Guo, A. Rowe, W. He, H. Pei, L. Qiang, and Y. Li. It service infrastructure for integrative systems biology. In *IEEE SCC 2004: IEEE Conference on Service Computing*, Shanghai, China, September 2004. `http://csdl.computer.org/comp/proceedings/scc/2004/2225/00/22250123abs.htm`.
3. Office for National Statistics (ONS). Guidelines for measuring statistical quality. Available at `http://www.statistics.gov.uk/StatBase/Product.asp?vlnk=13578`, 2006.
4. I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, July 1998.
5. I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 2002.
6. M. Ghanem, N. Giannadakis, Y. Guo, and A. Rowe. Dynamic information integration for e-science. In *UK e-Science All Hands Meeting*, Sheffield, UK, September 2002.
7. Groovy. `http://groovy.codehaus.org/`.
8. Discovery Net. `http://www.discovery-on-the.net/`.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Incorporating Theory Data into the Virtual Observatory

L.D. Shaw[1], N.A. Walton [1]
[1]Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge, CB3 0HA

Abstract

We describe work investigating how astronomical simulation data can be effectively incorporated into the Virtual Observatory. We focus specifically on determining whether the data query, access and retrieval standards being developed by the International Virtual Observatory Alliance for observational data can be similarly applied to, and fully support, the requirements of data extracted from astrophysical simulations. We present a data model for Simulation data and identify the extensions required to the Universal Content Descriptor astronomy metadata vocabulary to encompass simulation specific concepts and quantities. We also describe initial work on a new standard protocol to enable uniform access to simulation datasets.

1. Introduction

Over the last decade, the way in which we do observational astronomy has started to change. The slow, independant and uncoordinated manner in which data was accumulated in the past, through individual and unrelated observing programs, has been replaced by systematic and methodical projects aiming to map the sky in unprecedented detail. Advances in telescope, detector and computer technology have enabled us to explore the universe in a systematic and detailed way, in multiple wave-bands and at rapidly improving resolution. Individual ground and space based survey telescopes are currently producing tera-bytes of data. However, these instruments are merely precursors of those currently being designed and built. In direct correspondance to Moore's law, the rate at which astronomical data is collected doubles roughly every year and a half.

Consequently, Astronomy faces a data avalanche, and the astronomical community is confronted with a challenge: how can this huge flood of data be exploited to its maximum potential? How can we federate the disjointed archives of survey data and the vast numbers of small data-sets from individual observations and enable access to them through a uniform inferface? A solution to these challenges would provide a new and powerfull tool with which to probe the universe.

It was with these challenges in mind that the concept of a Virtual Observatory (VO) was conceived. The VO is a system in which the vast astronomical archives and databases around the world, together with analysis tools and computational services, are linked together into an integrated service. In reality, a number of national virtual observatories around the world are being developed concurrently (in the UK, Astrogrid [1]), each dealing with the data archives of their host astronomical community. In order to ensure that the Virtual Observatories around the world are able to interoperate, an international body was formed in 2001 – the International Virtual Observatory Alliance (IVOA) – to determine the standards to which individual virtual observatories must adhere to. Over the past few years, significant progress have been made by the IVOA in developing standards and protocols for astronomical data storage, discovery and retrieval.

Observational astronomy is not the only area of the field undergoing a data revolution. With the advent of parallel and grid computing and rapidly improving hardware, computational techniques in astrophysics and cosmology have become important tools in evaluating highly complex systems, from stellar evolution to the formation of galaxies and clusters of galaxies. However, there is currently no support for sim-

ulation data and services within the framework of VO standards. In this paper, we describe initial work investigating the new protocols and the changes to existing standards required to enable the incorporation of simulated datasets into the VO.

2. Standards for Simulation Data

Simulations are frequently used today in all areas of astronomy; from the birth, evolution and death of stars and planetary systems, to the formation of galaxies in which they reside and the formation of large scale structures, dark matter halos, in which galaxies themselves are thought to form. There is much variety in the processes being investigated and the underlying physics that govern them. Many different approaches have been chosen to tackle each problem, often employing very different algorithms to deal with the complex physics involved. There is clearly a huge amount of information that must be recorded in order to fully describe a simulation and its results, not all of which can be quantified in numerical terms. In order for simulated data to be included in the Virtual Observatory, we must first clearly identify all the different components that describe a simulated dataset. This is the purpose of defining an **abstract data model** for simulations (see Sec. 2.1).

At the IVOA interoperability meeting in Kyoto, the Theory Interest Group - charged with ensuring that the VO standards also meet the requirements of theoretical (or simulated) data - outlined a set of near-term targets with the aim of identifying where the existing IVOA standards and implentations must be updated to allow the discovery, exchange and analysis of simulated data. Computational cosmology is one example of an area of astronomy that should be a major benficiary of an international virtual observatory. Many independant groups are working towards solving the problems of hierarchical structure formation, performing large-scale simulations as an integral part of their investigations.

However, the results of many of these simulations are not publicly available. Furthermore, of those that are, the data is stored in a wide variety of (mostly undocumented) formats and systems, often chosen having been convenient at the time. Therefore the interchange and direct comparison of results by independant groups is uncommon as often much effort is required in obtaining, understanding and translating data in order for it to be of any use. A primary goal of the IVOA's Theory Interest group is thus to decide upon a standard file format for raw simulation data and a metadata language (based on the Universal Content Descriptors used for observational data [4]) with which to describe the contents. Based on this, a set of requirements of the standards being developed by the Data Access Layer (DAL) and Virtual Observatory Query Language (VOQL) working groups within the IVOA have been identified so that simulation data can seamlessly be discovered and retrieved through VO portals.

In this section, we describe the key standards for data discovery, querying and retrieval that have been developed and approved by the IVOA, and discuss whether, in their current form, they fully support the incorporation of simulated data in the VO, as outlined above. Although the standards discussed here do not cover the full range of those that are being developed by the IVOA, they are those that are most relevant to the differences between observed and simulated data.

2.1. Data Modelling and Metadata

In order for simulated data to be included in the Virtual Observatory, we must first clearly identify all the different components that describe a simulated dataset. To this purpose, initial attempts have been made to construct an abstract data model for simulations. In Figure 1 we demonstrate the current iteration of the 'Simulation' data model. This model was developed using the corresponding data model defined for observational data, Observation [2], as our starting point, modifying it where necessary. It is hoped that this will ensure that Simulation has a similar overall structure to Observation, differing only in the detail. There are two purposes to this approach. Firstly, it is hoped that a similarity between data models will aid the process of comparing simulated and observed datasets. Secondly, it maintains the possibility of defining an overall data model for astronomical data, real or synthetic.

A simulation can essentially be broken down into three main categories: Observation Data, Characterisation and Provenance. Observation Data describes the units and dimension of the data. It inherits from the Quantity data model (currently in development [3]) which assigns the units and metadata to either single or arrays of values. Characterisation describes not only the ranges over which each measured quantity

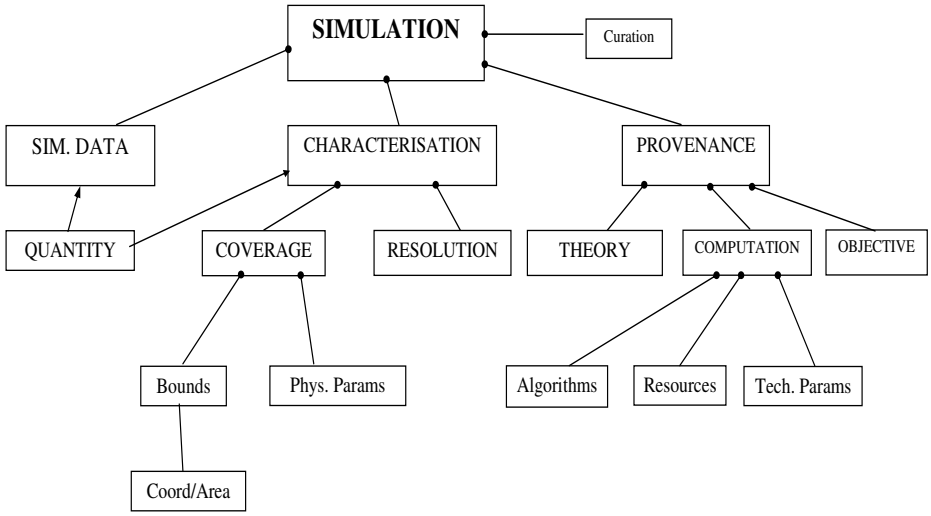


Figure 1: The principle components of the Simulation data model (see text)

is valid, but also how precise and how accurate they are. It is composed of Coverage and Resolution. These represent the different parameters constraining the data. Resolution describes the scales at which it is believed the simulation results begin to become significantly influenced by errors due to numerical effects, for example, due to the finite size and mass of simulation gridpoints or particles. Coverage describes the area of the Characterisation parameter space that the simulation occupies. It is itself composed of Bounds, which describes the range of values occupied by the simulation data, and PhysicalParameters, which consists of the set of physical constants that define 'the Universe' occupied by the simulation.

The third major component of Simulation is Provenance, which describes how the data was created. It consists of Theory, Computation and Objective. Theory represents a description of the underlying physical laws in the simulation. It is expected to consist of a reference to a publication or resource describing the simulation. Computation describes the technical aspect of the simulation and has three components – Algorithms, TechnicalParameters and Resources. Algorithm describes the sequence of numerical techniques used to evolve the simulation from one state to the next. It is expected that this also will contain a reference to a published paper or resource. TechnicalParameters are quantities representing the inputs to the algorithms, such as 'number of particles' and 'box size'. Resources describe the specifications of the hardware on which the simulation was performed. Objective describes the overall purpose of the simulation – what was the purpose of the simulation? What were the phenomenon that is was performed to investigate?

2.2. Metadata: UCDs for Simulations

In order to describe the quantities and concepts that are being outlined in the data models and published in data archives, a restricted vocabulary – Universal Content Descriptors (UCD [4]) – is being developed and controlled by the IVOA. UCDs are not designed to allocate units or names to quantites, they are meant to describe *"what the unit is"*. The overal purpose of UCDs is to provide a standard means of describing astronomical quantities – whether it be the luminosity of a galaxy or the exposure time of the instrument with which is was observed – using a restricted vocabulary (to prevent the proliferation of words), whilst retaining the flexibility to enable precise, non-ambiguous descriptions of the vast range of quantities that occur in astronomical datasets. Their main goal is to ensure interoperability between heterogeneous datasets. If an astronomer is searching for catalogues containing a specific quantity, she can use the UCD for that quantity to locate all those that contain it, whether it was the main purpose of the observation or not.

UCDs consist of a string constructed by combining 'words' seperated by semi-colons from the controlled vocabulary. Individual words may be composed of several 'atoms' seperated

by a period (.). The order of these atoms induces a hierarchy, where the following atom is a specific instance of its predecessor. Sequences of words each representing a specific concept are then combined to provide a describe of what the actual quantity is. For example, stat.error;phot.mag;em.opt.V would refer to the *error* on the measurement of the *photometric magnitude* (brightness) of an object in the *V-band* of the optical region of the electromagnetic spectrum.

However, untill now the UCD tree has been defined to deal specifically with observational related quantities. Although there exists the flexibility to describe the physical quantities measured from simulations (as these are often the properties of the objects that were being simulated), it is not currently possible to describe the properties and parameters of the simulations themselves. This includes some input physical parameters (i.e. cosmological parameters) that define the theoretical context of the simulation, and technical parameters that define its size, scope and resolution (e.g number of particles, length of simulation box side, time/redshift of a simulation output). We have therefore proposed a new branch of the UCD tree to encompass computatational techniques in astronomy; 'comp'. This branch can be used to describe both astrophysical (and cosmological) simulations, and data reduction and post-processing algorithms for both simulation and observational data.

2.3. Simple Numerical Access Protocol

One of the key objectives of the Virtual Observatory is to provide uniform interfaces to the different forms of astronomical data. This is now being realised with the development of a standard means of retrieving astronomical images and spectra. We are now developing a prototype standard for retrieving raw simulation data from a variety of astronomical simulation repositories – Simple Numerical Access Protocol (SNAP). SNAP is designed to enable uniform access to 'raw' simulation data. This includes the retrieval of all the particles or grid points within the simulation box at a particular timestep (known as a 'snapshot'), a specified sub-volume of a simulation (all the particles/grid-points within a certain region), or data from a post-processed simulation. The latter typically includes catalogues of objects that have been identified within a larger simulation or within a suite of simulations.

Astrophysical and cosmological simulations clearly cover an enormous range of scales and processes. Consequently, there is no absolute spatial scale for simulations, in the way that there is for astronomical image data (position on the sky, distance from earth). The most basic function that a SNAP service must provide is direct access to the raw particle data (or grid state) from a simulation output via an access URL.

However, due do rapidly improving hardware and fast and efficient parallel codes, the resolution, and therefore the filesize of a single snapshot can be extremely large. For example, many cosmological simulations (e.g. [5]) now contain 1024^3 particles within the simulation box. The file containing their positions and velocities will then be at least 20GB per timestep. It is therefore important to stress that, contrary to the procedure for retrieving observational images and data, simulation data cannot be retrieved via http with some kind of encoding for the binaries (e.g. base64) since this is extremely expensive operation when large datasets are being handled. FTP, or ideally GridFTP, must be used to retrieve the higher resolution simulations.

The SNAP standard is currently in the process of being defined and agreed through the IVOA. It is expected that v1.0 will be in place after the September 2006 Interoperability meeting.

Acknowledgements

LDS is support by a PPARC e-science studentship.

References

[1] Astrogrid: www.astrogrid.org

[2] http://www.ivoa.net/internal/IVOA/IvoaDataModel/obs.v0.2.pdf

[3] http://www.ivoa.net/twiki/bin/view/IVOA/IVOADMQuantityWP

[4] http://www.ivoa.net/twiki/bin/view/IVOA/IvoaUCD

[5] Bode, P., Ostriker, J.P. 2003, ApJS, 145, 1

[6] http://www.ivoa.net/twiki/bin/view/IVOA/IvoaTheory

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Common Instrument Middleware Architecture: Extensions for the Australian e-Research Environment

Ian M. Atkinson,[*a] Douglas du Boulay,[b] Clinton Chee,[b] Kenneth Chiu,[c] Kia L. Huffman,[d] Donald F. McMullen,[d] Romain Quilici,[b] **Peter Turner**,[*b] and Mathew Wyatt[a]

[a]School of Information Technology, James Cook University, Townsville, Qld, Australia.
[b] School of Chemistry, The University of Sydney, Sydney, NSW, Australia.
[c] Computer Science, State University of New York (SUNY) at Binghamton, NY, USA.
[d] Pervasive Technologies Lab, Indiana University, Bloomington, IN, USA.

Abstract

The Common Instrument Middleware Architecture (CIMA) model is being adapted and extended for a remote instrument access system being developed as an Australian eScience project. Enhancements include the use of SRB federated Grid storage infrastructure, and the introduction of the Kepler workflow system to automate data management and provide facile extraction and generation of instrument and experimental metadata. The SRB infrastructure will in effect underpin a Grid enabled network of X-ray diffraction instruments. The CIMA model is being further extended to include instrument control, and is being embedded as a component in a feature rich GridSphere portal based system for remote access.

1. Introduction

1.1 The Australian Context

Australia is a large island continent with most of its population primarily residing in a relatively small number of coastal cities. Given the size of the country, its easy to appreciate that the provision of remote access services to resources, such as scientific instruments and their data, offers significant efficiency and cost benefits. Further gains may be provided by harnessing distributed resource technologies such as grid storage and compute resources, work-flow tools and web services. In order to explore this potential, we're developing a pilot network of X-ray diffraction instruments equipped with Grid enabled services for remote access.

The development network encompasses instruments at James Cook University, Monash University, the University of Queensland and the University of Sydney (Fig. 1). That is, geographically the sites span the east coast of Australia across a distance of over 3000 km.

The selection of X-ray diffraction instruments for a remote access development programme, emerges naturally from the observation that crystallography offers well defined work-flows and data structures, and utilises relatively common (if not standardised) instrument types. X-ray diffraction instruments are sufficiently expensive items that many institutions are not in a position to own and operate such equipment.

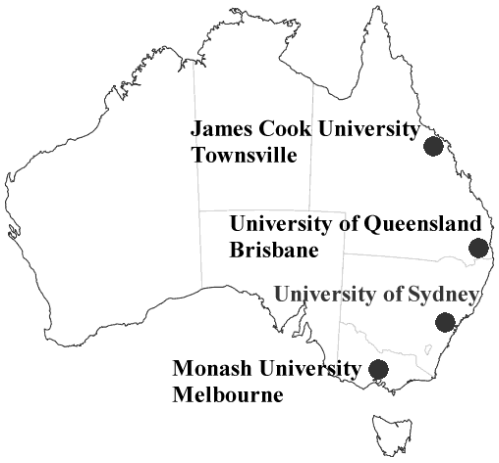


Figure 1. Remote Access Network Sites

1.2 Technology

The Common Instrument Middleware Architecture (CIMA)[1] project is developing a web services based API that embodies a generic or abstract representation of an instrument in terms of its type, identity, data and metadata output streams. The CIMA model provides a

middleware layer between the instrument and the network within which to standardize the representation of the instrument, to mediate access by downstream components, and to host extensions to the instrument's functionality. This middleware layer need not be co-located with the instrument, and could be elsewhere on the network. The CIMA model is intended to promote re-usability and hence reduce coding effort in changing instrument environments or eco-systems.

2. Adaptations and Extensions to CIMA

In close collaboration with the CIMA project team, we're adapting and extending the capabilities of CIMA in developing a rich GridSphere portal environment for remote instrument and data access.

2.1 Workflow and Data Management

Several significant changes to the CIMA model have been made with a view to increased flexibility and capability. The changes are depicted in Fig. 2 and are summarised as follows:

- NFS and MySQL data manager replaced with an SRB[2] and MCAT backend
- Use of Personal Grid Library (PGL)[4] for user friendly SRB data manipulation
- Moved from C++ and Perl to Kepler[3] to facilitate the handling of data management
- Ability to customise data storage via a visual workflow system
- Ability to create metadata schema definitions for experimental metadata
- Highly extensible, storage repository not restricted to SRB

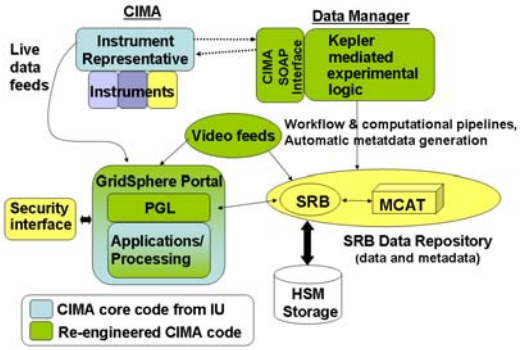


Figure 2. Extensions to the CIMA System

A significant goal of the project is to realise the possibility of instruments serving as first class members of the Grid. As Fig.3 suggests, Storage Resource Broker technology is being utilised to link the participating instrument sites.

At present the Data Manager for all sites is located at JCU with a production scale SRB storage facility. It is intended that each site will run its own Data Manager and SRB storage facility, which is to be federated across the sites into a single shared data space. The security and access rights issues associated with this store are presently being considered.

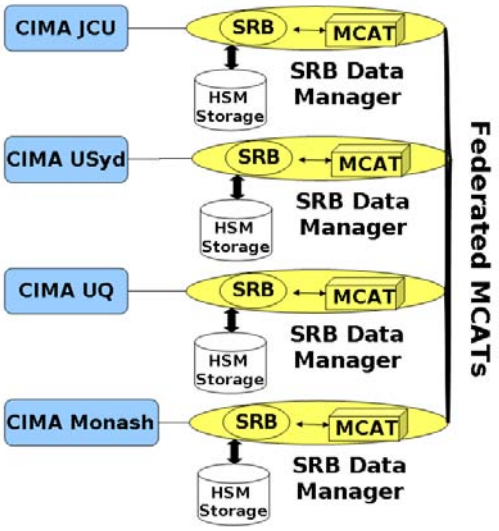


Figure 3. SRB Based Network Storage

The Personal Grid Library has been developed at James Cook University to provide easy web browsing and access to standalone or federated SRB instances. The PGL also supports annotation and the convenient display of object metadata. Annotation or metadata based searches can be undertaken, and stored images may be viewed.

As can be seen in Figure 2, Kepler plays a key mediating role in the work flow and data management. For use cases outside of X-Ray crystallography, the data management requirements will most likely be very different. Data management customisation for new applications would normally impose a heavy coding over-head. However, by exploiting Kepler workflows the development effort can be dramatically reduced. Using Kepler a customised data manager workflow can be configured in days, rather than weeks or months. Workflows in Kepler can be exported into XML, which can then be deployed to other instances of Kepler at different sites.

2.2 Remote Instrument Control

The remote desktop approach to remote instrument access, such as typified by the use of

VNC[5] (and its many variants), CITRIX[6], Tarantella[7] and NX[8], has the significant advantages of ease of set-up and familiarity. While convenient, these approaches are not ideal and can afford remote instrument users with excessive control over expensive and potentially dangerous instruments, and have a relatively poor security model. A significant disincentive in building custom-built remote access systems, is that there is a high coding overhead that may reproduce functionality already provided by an instrument manufacturer. A significant advantage of the custom built interface approach however, is that the actions of the remote instrument user can be tightly controlled, while at the same time services outside the desktop environment can be provided to offer a richer operating environment.

Accordingly a Gridsphere portal is being built that includes a portlet providing a purpose built interface to a diffractometer instrument that will tightly define the actions of a remote user, and so minimise accidental damage and injury. The portlet allows the user to enter data collection specifications and allows the execution of those instructions. Live views of the instrument and crystal sample are provided, together with the most recent diffraction image (see screen shot in Fig. 4.)

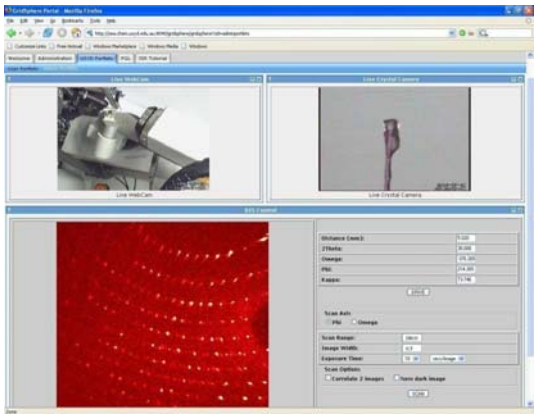


Figure 4. Instrument Control Portlet

A second portlet pane provides instrument status information, such as generator status, detector temperature and laboratory temperature.

A further portlet allows the user to browse the images collected to date (See Fig. 5), and includes a high speed 'flick through' capability. The browsers capability is to be further extended to include annotation and analysis.

Thus far CIMA has been developed solely for instrument and sensor monitoring. We are now extending the role of CIMA to include support for instrument control, and this involves including CIMA services as components in a modular portal architecture (see Fig. 6).

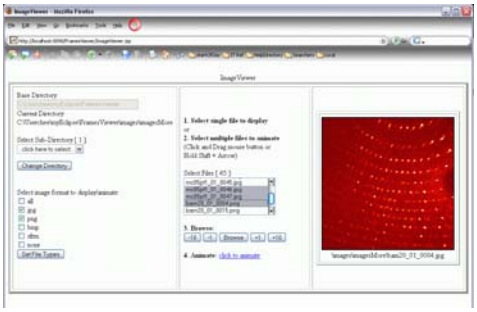


Figure 5. Portlet Diffraction Image Browser

A modular SOA model using Web Services has been adopted to provide maximum flexibility, with Web Services providing language and location independence. Container communication is based on SOAP messages containing XML parcels. A data cache minimises the need to seek (get) updated status information from the instrument.

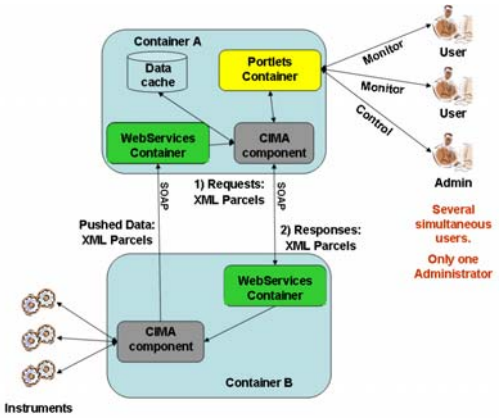


Figure 6. Instrument Control Architecture

The remote steering portlet is also being enhanced with a virtual depiction of the instrument, allowing a data collection simulation and assessment before execution. The instrument simulation utilises and extends the DS diffractometry simulation program.[9] The DS software has been ported from HP based C and OpenGL to Linux, Java, JOGLand OpenGL for this work, and can now be incorporated into the portlet system. Currently we're adapting the OpenGL graphics to X3D[10] for further flexibility and functionality. The DS implementation can reflect the current state of the diffractometer, and can simulate the axis and detector movements of a full data collection. Further work is being uncertaken to more precisely represent the instrument being simulated. The simulator has training benefits

and can help minimise the risk of damage to the instrument. The simulator also provides a back-up for a web camera failure and reduces the impact of the dark lab problem.

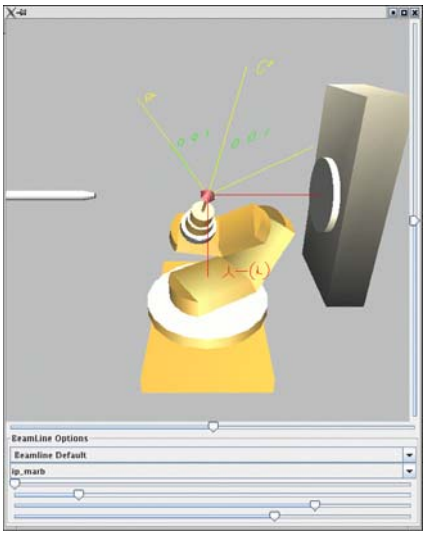


Figure 7. DS[9] Based Instrument Simulation

A significant challenge remaining is the incorporation of a single sign on security system, and currently we're exploring the use of Shibboleth. [11]

References

(1) (a) Huffman, K.L., Gupta, N., Devadithya, T., Ma, Y., Chiu, K., Huffman, J.C., Bramley, R., McMullen, D.F. Instrument Monitoring, Data Sharing and Archiving Using Common Instrument Middleware Architecture (CIMA): A new approach to modernize and standardize the data collection procedures in an X-ray Crystallography Laboratory. Journal of Chemical Information and Modeling, accepted February 27 2006. (b) CIMA: http://www.instrumentmiddleware.org
(2) Storage Resource Broker: http://www.sdsc.edu/srb
(3) Kepler workflow project: http://www.kepler-project.org
(4) Portable Grid Library: http://plone.jcu.edu.au/hpc/hpc-software/personal-grid-library/releases/0.3
(5) Virtual Network Computing; http://www.realvnc.com/ (also TightVNC, RealVNC, UltraVNC, and TridiaVNC)
(6) CITRIX: http://www.citrix.com
(7) Tarantella; now renamed Sun Secure Global Desktop: http://www.sun.com/software/products/sgd
(8) NoMachine (NX); http://www.nomachine.com/
(9) J. Appl. Cryst. (1995). 28, 225-227. DS - a 3D graphics simulation program for single-crystal diffractometer. Zheng, M. Yao and I. Tanaka.
(10) X3D: www.web3d.org
(11) SHIBBOLETH: shibboleth.internet2.org

Acknowledgements: The support of the Australian Department of Education, Science and Training (DEST), for the DART project (dart.edu.au), the Australian Research Council (ARC) eResearch Program, the ARC Molecular and Materials Structure Network (mmsn.net.au) and GrangeNet is gratefully acknowledged. Dr Min Yao is thanked for generously providing the DS source code, and the assistance of Dr Darren Spruce is also gratefully acknowledged.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

ShibVomGSite: A Framework for Providing Username and Password Support to GridSite with Attribute based Authorization using Shibboleth and VOMS

Joseph Olufemi Dada & Andrew McNab

School of Physics and Astronomy, University of Manchester, Manchester UK

Abstract

The GridSite relies on the Grid credentials for user authentication and authorization. It does not support username/password pair and attribute-based authorization. Since Public Key Infrastructure (PKI) certificate cannot be installed on all the devices a user will use, access to GridSite protected resources by users while attending conferences, at airport kiosk, working in Internet Café, on holidays etc. is not possible without exposing the certificate's private key on multiple locations. Exposing the private key on multiple locations poses a security risk. This paper describes our work in progress, called ShibVomGSite: a framework that integrates Shibboleth, VOMS and GridSite to solve this problem. ShibVomGSite issues to users a username and Time Limited Password bind to their Distinguished Name (DN) and Certificate Authority's DN in a database, which the users can later use to gain access to their attributes that are used for authorization.

1 Introduction

GridSite [1-3] is a certificate-based website management system that allows members of an organization to collaborate in maintaining web pages etc. It was originally developed for the management of GridPP project's web site [4]. Authentication is based on Grid credentials, but with unmodified web browsers such as Netscape and Internet Explorer. To access the GridSite protected resources, users must apply and obtain certificate from the Certificate Authority (CA). The certificate and private key protected by a pass phrase must be installed on every devices that users will use to access the GridSite protected resources. Installing certificate and its private key on multiple locations poses a security risk. This limitation prevents users having access to GridSite protected resources while attending conferences, at airport kiosk, working in Internet Café, on holidays etc. This paper describes a framework called ShibVomGSite; we developed to overcome this limitation.

Our framework provides a shibbolized access to GridSite resources. Users use a Time Limited Password associated with their Public Key Infrastructure (PKI) certificates [5] to gain access to the their attributes that are used for authorization. We developed the MyIdentity service to handle the issuing of username and Time Limited Password to users at their home institution. MyIdentity service also allows users to manage their identities in the database.

The GridPP collaboration involves a community of many particle physicists, computer scientists and site administrators with members located at UK universities and international laboratories. These various affiliations make it imperative to link Shibboleth Attribute Authority at the origin site with Virtual Organization Membership Service (VOMS) [6, 7] in order to get relevant member's VOMS-Attributes necessary for authorization. To achieve this, we developed the Voms Attribute Service for GridSite and Shibboleth (VASGS) that integrates VOMS with Shibboleth. Shibboleth uses VASGS to retrieve users attributes from VOMS, which it then passes together with other attributes to GridSite for authorization purposes.

For the authorization, we introduce the concept of GridSite Authorization Module for Shibboleth and Apache Server (GAMAS). GAMAS integrates with Shibboleth at the resource provider or target site. The rest of this paper is organised as follows: Section 2 briefly describes the GridSite authentication and

authorization process, VOMS and Shibboleth. ShibVomGSite system is described in section 3 along with a description of how its components work together to achieve the objective of our work. A brief description of the prototype of MyIdentity service, VASGS service and GAMAS is presented in section 4, and section 5 gives the conclusion and further work.

2 Background

In this section we present a brief description of authentication and authorization in GridSite and provide an overview of the two technologies that are relevant to our work: Shibboleth and VOMS. A detailed description of Shibboleth, VOMS and GridSite can be found on the individual websites [1, 6-8].

2.1 GridSite Authentication and Authorization

Mutual authentication in GridSite is established based on Grid credentials that require the use of X.509 identity certificates [5]. A user needs to have a valid X.509 certificate together with the corresponding private key in order to proof his/her identity to the GridSite resources.

After the proof of identity, the user needs to be authorized to gain access to the GridSite resources based on the resource provider access policy. The GridSite Apache module (mod_gridsite) implements authorization based on X.509, Grid Security Infrastructure (GSI) [9] and VOMS credentials. It uses GACL, the "Grid Access Control Language" [3] provided by the GridSite/GACL library. This allows access control to be specified in terms of attributes found in Grid Credentials. The Access Control Lists (ACLs) consist of a list of one or more Entry blocks. When a user's credentials are compared to the ACL, the permissions given to the user by Allow blocks are recorded, along with those forbidden by Deny blocks. When all entries have been evaluated, any forbidden permissions are removed from those granted.

2.2 Shibboleth

Shibboleth is standards-based, open source middleware software, which provides Web Single Sign On (SSO) across or within organizational boundaries. It allows sites to make informed authorization decisions for individual access of protected online resources in a privacy-preserving manner [4]. Shibboleth consists of two major software components: the Identity Provider (IdP) and Service Provider (SP). The two components are deployed separately but work together to provide secure access to Web-based resources. The operation of Shibboleth is based on the Security Assertion Markup Language (SAML) standard [10], published by the OASIS [11]. The principle behind SAML's design and Shibboleth's is federated identity. Federated identity technology permits organization with disparate authentication and authorization methods to interoperate thereby extending the capability of each organization's existing services rather than replacing them.

Shibboleth exchanges attributes across domains for authorization purposes. Its architecture is dependent on PKI, which it uses to build trust relationship between the several Shibboleth components of the Federation members. Figure 1 shows the authentication and authorization process in Shibboleth. As shown in the figure, an IdP normally located at the origin site identifies the users while SP at the target site protects the resources. When a user accesses a shibbolized resource at the target site for the first time, the Shibboleth Indexical Reference Establisher (SHIRE) directs the user to go to Where Are You From (WAYF) to pick his/her domain (origin site). The user's browser is then redirected to the authentication server at the origin site for user authentication. After the user is authenticated, the browser is redirected back to the target site along with the user's handle and authentication assertion. The authentication assertion is a proof to the target site that the user has been successfully authenticated. The Shibboleth Attribute Requester (SHAR) at the target site uses the user's handle to request for attributes of the user from the Attribute Authority (AA) at the origin site. The attributes are then passed to the Shibboleth Authorization (ShibAuthZ) module, which will make an access control decision based on these attributes. GAMAS takes over the function of ShibAuthZ in ShibVomGSite system.

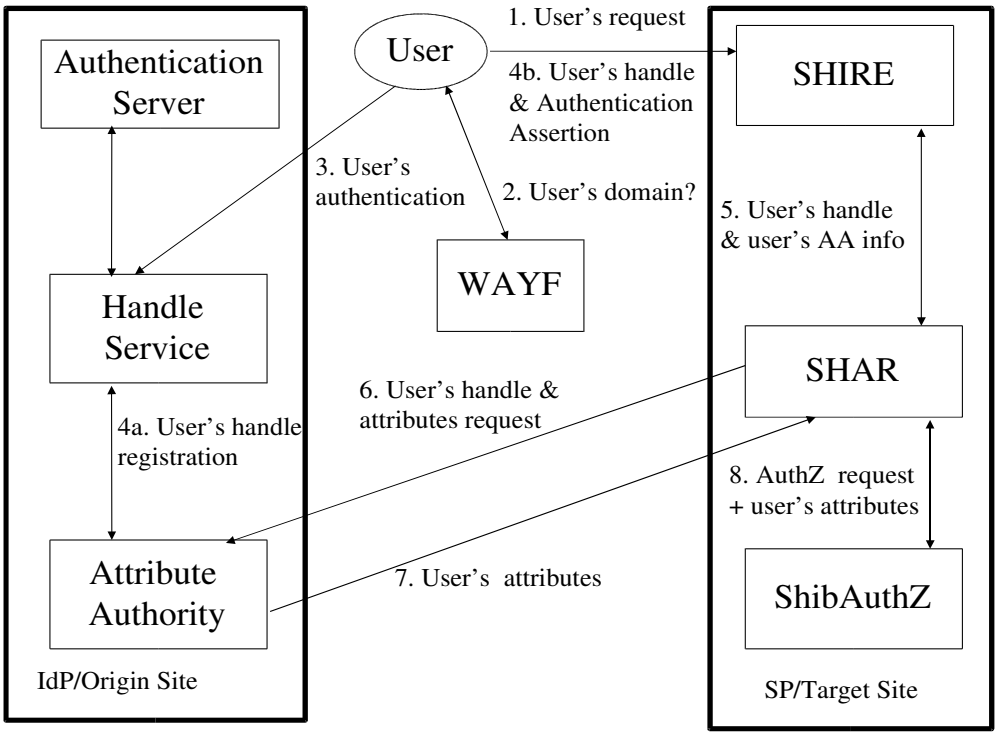


Figure 1: Architecture of Shibboleth

2.3 Virtual Organization Membership Service (VOMS)

Virtual Organization Membership Service provides information on the user's relationship with her Virtual Organization: her groups, roles and capabilities. The service is basically a simple account database, which serves the information in a special format (VOMS credential). The VO manager can administrate it remotely using command line tools or a web interface. An authenticated user (or any principal) can request membership of a VO, and request group membership, role, and capability entitlements [6].

Once the user has been granted the appropriate VO membership and attributes within a VO, he may request a short-lived credential. The user runs voms-proxy-init with optional arguments relating to which VOs, groups, roles, and capabilities he wishes for his current credential. VOMS issues a short-lived Attribute Certificate [12] to the authenticated user, which the user may then present to resources on the Grid.

However, its present implementation doesn't support issuing of user's attributes or authorization data to a third party on behalf of the user. We have developed a web service that can be plugged into VOMS to enable a trusted third party (e.g. Shibboleth IdP) requests user's attributes on behalf of the user, which are then passed to the Service Provider for authorization purposes.

3 ShibVomGSite System

In this section, we present the ShibVomGSite system that addresses the problems enumerated in the introduction. ShibVomGSite consists of three major components that integrate Shibboleth, VOMS and GridSite to enable GridSite supports username/password and attribute-based authorization: MyIdentity Service, VASGS service, and GAMAS. We describe these components in the sub sections that follow. Figure 2 shows these components, their interactions and how they interact with Shibboleth to achieve the objective of this work. The steps shown in the figure are explained below:

1. The user contacts Shibboleth/GridSite protected resource site with a browser, requesting to access a Shibboleth-GAMAS protected target service.
2. The user is redirected to the IdP for authentication.
3. IdP calls the MyIdentity service for user authentication.

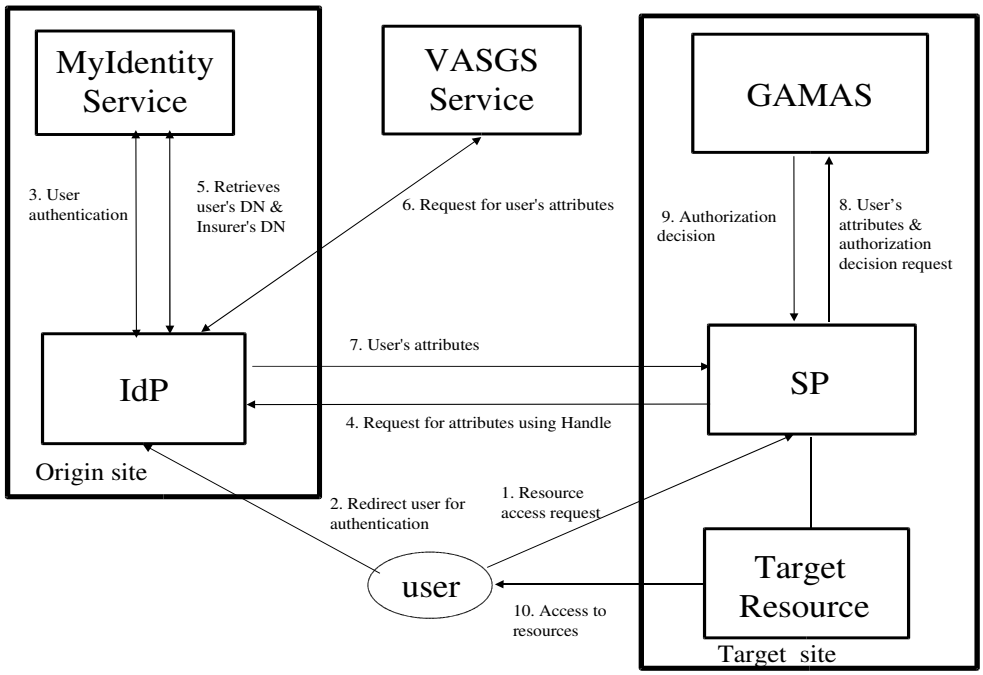


Figure 2: ShibVomGSite Architecture

4. After successful authentication, the browser is redirected to the SP together with handle. The SP at the target site gets the handle and sends the handle to the IdP of the origin site for attributes query.
5. The IdP retrieves the user DN and CA's DN from MyIdentityDB.
6. The IdP authenticates to the VASGS-Service using host PKI certificate, and uses user's DN and CA's DN as parameters to request for user's attributes from VOMS through the VASGS service. The VASGS service returns the user's attributes to the IdP.
7. The IdP sends the attributes together with the user's DN if required back to the SP.
8. The SP uses the user's attributes to request for authorization decision from GAMAS.
9. GAMAS carries out authorization process and passes its decision back to SP.
10. SP grants or denies access to the Target Resource depending on the authorization decision from GAMAS.

3.1 **MyIdentity Service**

MyIdentity service carries out authentication of users using certificate and enables them to manage their username and password bind to their DN. It is simply a database with an interface developed in C. Unlike MyProxy [13] that stores proxy credentials, MyIdentity only stores users' DN and CA's DN bind to the username/password, which are later used by Shibboleth to retrieve user's attributes from VOMS server. The components of the MyIdentity service are shown in Figure 3.

The first operation a user must perform in order to get access to GridSite resources without using certificate is to request for a username and Time Limited Password from the MyIdentity Service. The steps involve are described below:

1. The user and the MyIdentity Service authenticate each other with their certificates using HTTPS protocol.
2. MyIdentity service extracts the user's DN and CA's DN from his/her certificate and issues a username and time limited password to the user through the same web browser the user used for the authentication. User can change his/her password immediately or within 6 days.
3. MyIdentity service saves the username and password in encryption form together with the DN and CA's DN in database.

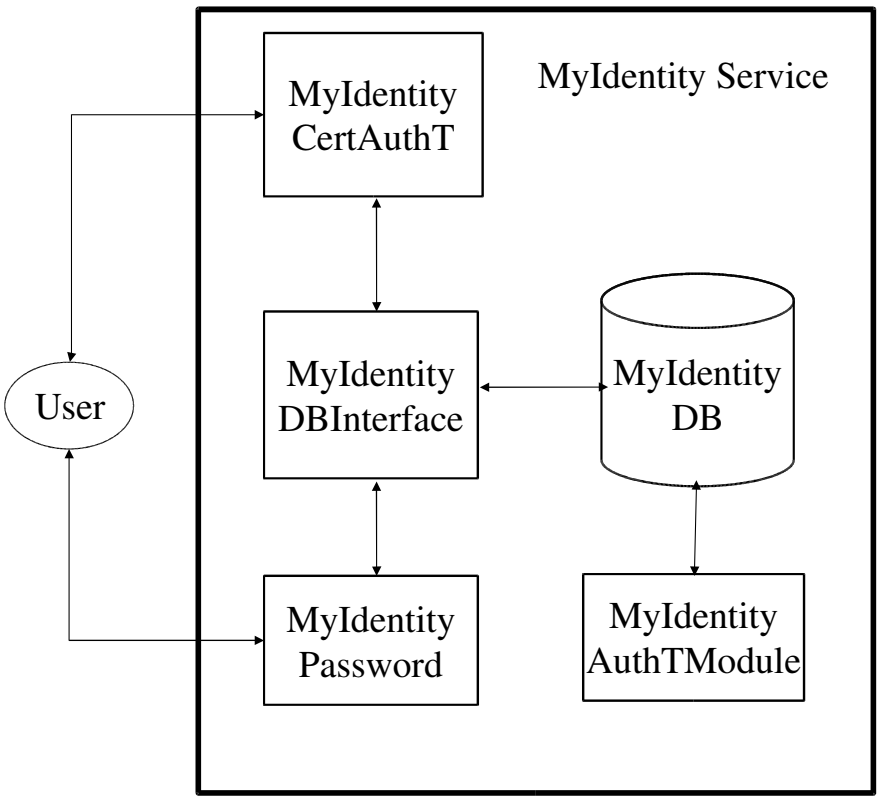


Figure 3: User interaction with MyIdentity Service

The user has the opportunity of managing his/her information in the MyIdentityDB using the MyIdentityPassword module.

Shibboleth IdP uses MyIdentityAuthT Apache module for the authentication of users anytime users attempt to access GridSite protected resources on the target site. MyIdentityAuthT Apache module is based on Apache module (mod_auth_mysql) [14]. The module is integrated into the MyIdentityDB that contains the username/password and other user's information. A full paper on MyIdentity service is in preparation.

3.2 Voms Attribute Service for GridSite and Shibboleth (VASGS)

VASGS service is made up of two components: VASGS-VOMAttribute Web Service and VASGS-ConnectorPlugIn for IdP. Figure 4 shows the interaction between Shibboleth IdP and VASGS service. IdP connects VASGS-VOMAttribute Web Service to get user's attributes (groups, roles, capabilities etc.) from VOMS server using VASGS-ConnectorPlugIn. The attributes are combined with the others, which are then pass to the SP for authorization. The advantage of VASGS Service is that, users don't need to apply for Attribute Certificate to access GridSite resources; attributes are pull directly from the VOMS. Our VASGS service therefore allows IdP to use VOMS as an Attribute Repository.

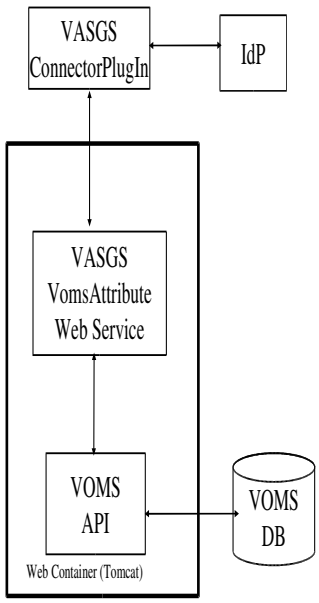


Figure 4: Structure of VASGS Service

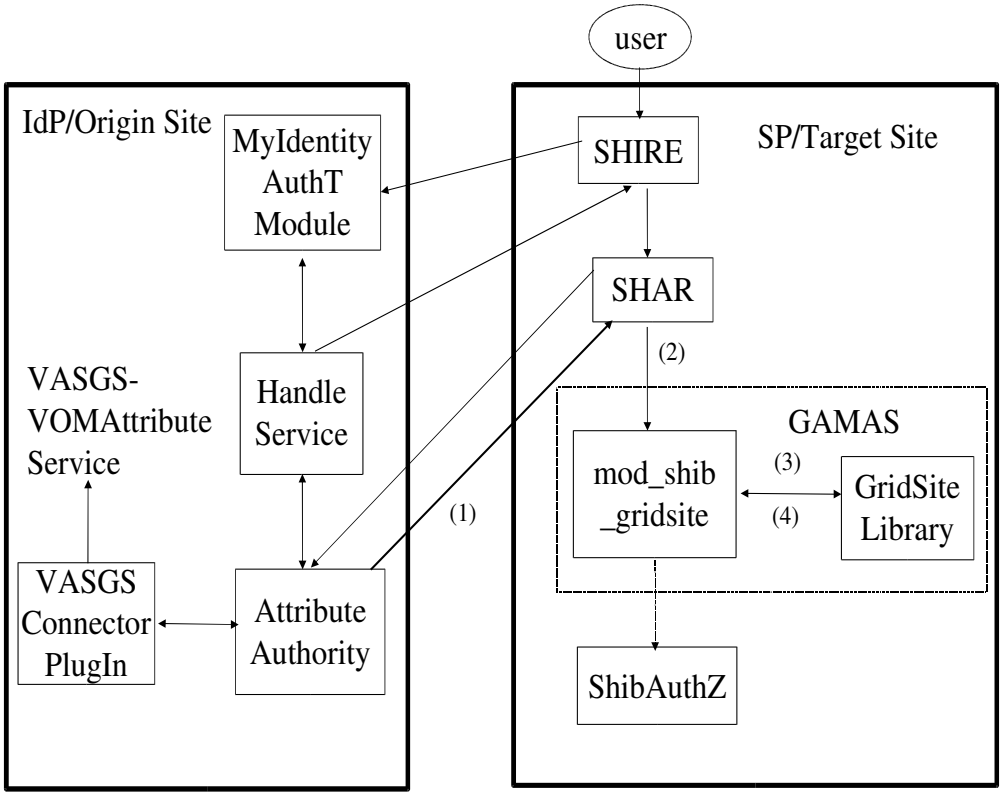


Figure 5: Integration of GAMAS with Service Provider

3.3 GridSite Authorization Module for Shibboleth and Apache Server (GAMAS)

Figure 5 shows the structure of GAMAS and how it integrates with SP to carry out the authorization process. The mod_shib_gridsite is the core module of GAMAS. It interfaces with Shibboleth and Apache server to collect all the attributes necessary for making authorization decision and passing these attributes to the GridSite/GACL/XACML library. Authorization decision is passed back to the mod_shib_gridsite, which translates it to "OK" or "HTTP_UNAUTHORIZED" error codes. Apache will either send the requested resource or a page with the error information back to the user web browser depending on the result. Since GAMAS returns a definite result when mod_shib_gridsite is active, Shibboleth authorization module (ShibAuthZ) is not invoked.

The mod_shib_gridsite must appear before the Shibboleth Apache module (mod_shib) in the Apache 2.0 configuration file. Since each location in Apache configuration file may use a different form of authorization, GAMAS is only active if the "GridSiteAuth" directive is present for the location. If it's not present, mod_shib_gridsite will return DECLINED, so that Shibboleth or any configured authorization module will be invoked.

To explain how GAMAS integrates with SP at the target site, the interactions between SP and GAMAS during the authorization phase (shown with numbers in Figure 5) are explained as follows:

1. The authorization phase begins after SHAR component of the SP successfully received user's attributes from Attribute Authority component of the IdP as earlier explained in section 3. In this phase, mod_shib_gridsite is invoked first by the Apache server.
2. If requested location is not being protected by GAMAS, the mod_shib_gridsite will return DECLINED and the Shibboleth authorization function ShibAuthZ or any other authorization function for the location will be invoked, otherwise the user's attributes and DN are acquired by the mod_shib_gridsite from the HTTP headers.
3. Mod_shib_gridsite calls the gridsite/GACL/XACML library to make an authorization decision, which is based on user's attributes and DN.

4. After the decision, the granted/denied decision is returned back to mod_shib_gridsite.

5. Mod_shib_gridsite returns the decision to Apache server. The user is then granted or denied access to the target resource according to the result of the decision.

4 Prototype Implementation

We have implemented the prototype of MyIdentity service, VASGS service and GAMAS. In this section, we briefly describe our implementation.

MyIdentity service prototype is implemented in C. The database server used is MySQL. Users can login with their certificates and obtain username and password. MyIdentityCertAuthT module is a Common Gateway Interface (CGI) script that connects to the MyIdentityDB using MyIdentityDBInterface. MyIdentityPassword, which allows users to manage their records, is also a CGI script that uses MyIdentityDBInterface to interact with the MyIdentityDB. MyIdentityAuthT module is the authentication server for the Shibboleth, which is based on the Apache module: mod_auth_mysql [14].

VASGS service is implemented in JAVA. It is based on Web Service technology. It has two sub components as earlier described: VASGS-VOMAttributeService that resides with VOMS server and the VASGS-ConnectorPlugIn that Shibboleth uses to invoke the VASGS-VOMAttributeService on the VOMS server. VOMAttributeService runs inside a Tomcat web container just like the others services provided by the VOMS server while ConnectorPlugIn is a JAVA class that Shibboleth invokes to connect to the VOMAttributeService.

GAMAS (mod_shib_gridsite and gridsite library) is implemented in C as Apache module. It is an extension of the mod_gridsite. As earlier described, it receives users' attributes from Shibboleth to make authorization decision.

5 Conclusion and Further Work

We have presented a ShibVomGSite framework that provides username/password support and attribute-based authorization to GridSite. This framework allows users access to GridSite protected resources anywhere and anytime using time limited password.

MyIdentity service binds user's DN and CA's DN with the username and Time Limited Password in a database (MyIdentityDB). It also serves as an attribute repository for the IdP, and provides DN and CA's DN used as parameter to obtain VOMS attributes for users with the help of the VASGS service. We also described GAMAS that uses the VOMS attributes received from Shibboleth for authorization. GAMAS receives the attributes through the Shibboleth SP (mod_shib), carries out authorization process and returns decision result to the Apache Server, which grants or denies user's request depending on the result of the authorization decision.

The work presented in this paper is a work in progress. Efforts are continue to further develop the existing prototype to a full working system suitable for deployment. We are also working on integrating ShibVomGSite system with Flexible Access Middleware Extension (FAME) [15].

6 Acknowledgements

This work was funded by the Particle and Astronomy Research Council through their GridPP programme.

Our thanks also go to other members of the various EDG and EGEE security working groups for providing much of the wider environments for this work.

7 References

[1] Grid Security for the Grid, Web Platform for Gird, http://www.gridsite.org/ .

[2] McNab, A., The GridSite Web/Grid Security System, Software Practice. Exper., http://www.interscience.wiley.com, 35:827-834, 2005.

[3] McNab, A., "Grid-Based Access Control and User Management for Unix Environments, File systems, Web Sites and Virtual Organizations", in Proceedings of CHEP 2003, La Jolla, CA, 2003.

[4] UK Computing for Particle Physics, http://www.gridpp.ac.uk/.

[5] Houseley, R., Polk, W., Ford, W. and Solo, D., Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, IETF, 2002.

[6] Alfieri, R., Cechini, R., Ciaschini, V., Spataro, F., dell' Agnello, L., Frohner, A. and Lörentey, K., From gridmap-file to VOMS: managing authorization in a Grid environment,

http://www.cnaf.infn.it/~ferrari/seminari/griglie05/lezione02/voms-FGCS.pdf, 2004.

[7] EDG-VOM-ADMIN Developer Guide, http://edgwp2.web.cern.ch/edgwp2/security/voms/edg-voms-admin-dev-guide.pdf.

[8] Shibboleth Project, Internet2, http://shibboleth.internet2.edu/.

[9] Welch, V., Siebenlist, F., Foster, I., Bresnahan, J., Czajkowski, K., Gawor, J., Kesselman, C., Meder, S., Pearlman, L. and Tuecke, S. Security for Grid Services. In International Symposium High Performance Distributed Computing, 2003.

[10] Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)V1.1, http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf.

[11] OASIS, http://www.oasis-open.org/.

[12] Ciaschini, V., A VOMS Attribute Certificate Profile for authorization, http://infnforge.cnaf.infn.it/docman/view.php/7/58/AC-RFC.pdf, 2004

[13] Novotny, J., Tuecke, J. and Welch, V., "An Online Credential Repository for the Grid: MyProxy", http://www.globus.org/alliance/publications/papers/myproxy.pdf

[14] Mod_auth_mysql: http://modauthmysql.sourceforge.net/.

[15] FAME-PERMIS - Flexible Access Middleware Extension to PERMIS, http://www.fame-permis.org/.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Instance-Level Security Management in Web Service Business Processes

Dacheng Zhang
University of Leeds
dcz@comp.leeds.ac.uk

Jianxin Li, Jinpeng Huai
Beihang University
{lijx, huaijp}@act.buaa.edu.cn

Abstract

By using Web services, people can generate flexible business processes whose activities are scattered across different organizations, with the services carrying out the activities bound at run-time. We refer to an execution of a Web service based automatic business process as a business session (multi-party session). A business session consists of multiple Web service instances which are called session partners. Here, we refer to a Web service instance as being a stateful execution of the Web service. In [8], we investigate the security issues related to business sessions, and demonstrate that security mechanisms are needed at the instance level to help session partners generate a reasonable trust relationship. To achieve this objective, an instance-level authentication mechanism is proposed. Experimental systems are integrated with both the GT4 and CROWN Grid infrastructures, and comprehensive experimentation is conducted to evaluate our authentication mechanism. Additionally, we design a policy-based authorization mechanism based on our instance-level authentication mechanism to further support trustworthy and flexible collaboration among session partners involved in the same business session. This mechanism allows an instance invoker to dynamically assign fine-grained access control policies for the new invoked instance so as to grant other session partners the necessary permissions.

1. Introduction

A business process is a collection of related structured activities undertaken by organizations in pursuit of certain business goals. When considering the automatic generation of business processes whose activities and data are separated by organisational boundaries, the heterogeneity of software components and legacy systems is an essential issue, and needs to be carefully investigated. In the past decades, much research has been put in identifying techniques and methodologies to effectively integrate heterogeneous software components and reuse legacy software systems; Web services are the latest attempts in this research field. Basically, Web services are an evolution of traditional Web applications. Through the use of Web service technologies, people can create complex applications at run time by integrating loosely coupled, heterogeneous, reusable software components. However, the flexibility and the peer-to-peer style of Web service business processes also introduce new challenges to security systems. For instance, two competitive business sessions may have activities undertaken by instances spawned by the same Web service. Although these instances are potentially competitive, they are executed on the same computer and share same resources, e.g. data, memory, etc. Moreover, because the session partners of a business session may be controlled by different organizations, the business session management system cannot monitor and manage its session partners as effectively as in the conventional business process instances. All these issues make business sessions vulnerable to malicious attacks. In this paper we present our efforts in generating security boundaries for business sessions. Our approach achieves authentication and authorization for business sessions at the instance level. The authentication system can generate and verify the identities of session partners at run-time. In comparison with conventional authorization approaches for business processes, our authorization system is more flexible as it permits session partners to dynamically assign the access control policies to the instance they invoked. Moreover, with the assistance of our authentication system, the authorization system can enforce the business session constraint which specifies that an instance can get the access to another instance only when the two instances are involved within the same multi-party session. In Section 2, we discuss instance-level authentication issues for web services and introduce our authentication system. In Section 3, we show our experimental systems which are used to evaluate the overhead introduced by the authentication

system. Experimental results are also presented and analysed. In Section 4, we briefly discuss the limitations of existing authorization systems in business process management, and then introduce our instance-level authorization system. Finally, Section 5 concludes the paper.

2. Instance-Level Authentication

Authentication is always a fundamental concern when a security infrastructure for distributed systems is designed. In order to validate the identities of the principals in distributed systems, authentication systems are employed. Without losing generality we regard an automatic business process as an automatic distributed system, and the principals working within a business session are Web service instances. In this section, we discuss the authentication issues for Web service instances within business sessions and propose an instance-level authentication mechanism.

2.1 Authentication Issues in Web Service Business Sessions

When a Web service receives an initial request from a business session, it normally generates a new instance to deal with this request. This instance is then involved within the business session and may also deal with other requests from its session partners. Consider that the session partners may be generated and bound at run-time, and cooperate in a peer-to-peer way [1]. Moreover, the specification of the communication amongst the session partners on some occasions cannot be precisely defined in advance. Thus, there are cases where a service instance has to deal with requests from other instances which it never communicated before [8]. When an instance receives such a request, the instance needs an authentication mechanism to help it verify the identity of the sender in case that the request is sent from an instance in other business session by mistake, or from a malicious attacker.

Essentially, there are two goals for an authentication system [7]. One is to establish the identities of the principals; another is to distribute secret keys for principals. Such secret keys can be used in further communication amongst the principals or used to generated new short-term secrets. In conventional authentication mechanisms (e.g., PKI, Kerberos, etc.), the identities of the users and the secret (e.g., password) used in the authentication processes are generated and deployed in advance. For example, in PKI, the certificate authority can generate a public key pair and sign a certificate for every user. The public key pair is forwarded to the user out-of-band, and the certificate is stored in a directory server and published over the network. Thus users can use the certificates and the key pairs obtained in advance to prove their identities during the communication with other users. In contrast to these classical authentication mechanisms, instance-level authentication systems have to generate and distribute both the identities of the instances and the secrets used to prove the identities at run-time as the instances are dynamically generated.

Most current Web service systems have not carefully considered authentication issues in business sessions yet. In GT4, service instances can be identified by their resources keys, but the solutions to proving the possession of the identifiers are not standardized. We can use user certificates to generate proxy credentials at run-time for every new generated instance, and thus verify the identities of service instances. However, the proxy certificate solution cannot be directly used to enforce the security boundaries of business sessions, as there is no notion of business session in this solution at all.

2.2 Instance-Level Authentication in [5]

Hada and Maruyama [5] propose a session authentication mechanism that can realize basic authentication functions for session partners. In their protocol a Session Authority (SA) component is introduced to take charge of distributing session authentication messages. The SA generates a session secret for every business session. When a new service instance is invoked and involved within a business session, the associated session secret will be forwarded to the instance, and thus the instance can use this secret to authenticate itself to its session partners. This authentication protocol can effectively authenticate the session partners working within the same business session and prevent the uncontrolled messages transport across the boundaries of business sessions. However, there are still many security issues in the area of business session management which needs to be carefully considered. For instance, Hada and Maruyama's protocol only can distinguish the session partners in one business session from the service instances in other business sessions. However, in some scenarios a fine-grained control over the session partners is required. For example, a Web service may have

multiple instances acting in different roles within the same business session; these instances need to be explicitly distinguished. Moreover, in Hada and Maruyama's protocol, there is only one session secret for every business session; the whole business session will be comprised if this session secret is disclosed. Thus, this protocol is not suitable for business sessions with high security requirements. Based on above discussion, we propose in [8] a fine-grained authentication protocol which is complementary to Hada and Maruyama's protocol. Our protocol explicitly identifies the session partners at the instance level. We also design a new key management mechanism which generates secrets and distributes them to session partners at run time. These secrets can be used in the authentication processes amongst session partners.

2.3 Multi-Party Authentication System for Service Instances

Since Web service instances are the principals of business sessions. Our authentication system attempts to directly authenticate service instances rather than users, which is different from the conventional authentication systems mentioned above. In our authentication system, every instance is associated with a pair of keys generated by using the Diffie-Hellman algorithm. The public key is used as the identifier of the instance, while the private key is kept secretly and used to prove the possession of the identifier. Similar to the solution in [5], there is a SA in our solution. All the identifiers of the session partners and other related information of a business session are stored in the SA. When a new service instance is invoked by a business session, the identifier and other related information will be sent to the SA so as to guarantee the validity of the information kept by the SA. A SA instance is associated with this business session to manage the associated session information. Additionally, the SA instance can provide reliable real-time information to session partners. For example, when an instance receives a request from a stranger, the instance can query the SA instance whether the requester is a session partner.

As mentioned above, an important objective of authentication systems is to generate and distribute secret keys for principals. Because our authentication system is an identity-based mechanism, if two session partners have obtained each other's identifiers they can generate and share a secret key without any additional communication. In our authentication protocol, Message Authentication Codes (MACs) are used to prove the integrity of the messages transported between session partners and thus verify the origin of a message. A nonce is inserted into the SOAP header of every message to foil the man-in-the-middle attack before messages are sent out. The nonce is generated from a unilaterally increased counter which is held by each Web service instance. When sending a message out, the instance increases its counter. Suppose instance *A* receives a message from instance *B*. If the counter number in the message is smaller than or equal to the counter number in the message sent from *A* before, *B* will reject this message. Interested reader can be referred to [8] for more detailed introduction about our authentication protocols.

3. Experimental System

Figure 1 illustrates the process of invoking a new Web service instance and introducing it to the SA as a session partner. Because the authentication system is implemented on the Grid infrastructures, a Web service is associated with a factory service to manage the resources where the state information is stored. In our experimental system, the identifier of an instance and associated private key is stored within a resource, and the instance identifier is identical to the resource key. In the first step (messages 1 and 2), the invoking instance on service 1 contacts factory service 2 in order to generate a new resource.

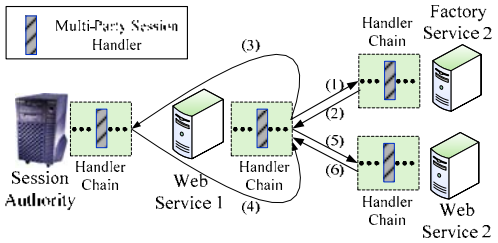


Figure 1: Operations of authentication

In Grid, the state information of Web services is stored in resources. Therefore, we can locate a service instance when the corresponding resource is found. After receiving the reply from factory service 2 and gaining the resource identifier, the invoking instance introduces the new invoked instance to the session authority (messages 3 and 4). If the reply from the SA implies that the new invoked instance has been accepted as a session partner, the invoking instance sends message (message 5) to the new

invoked instance and waits for the response (message 6). Compared with the original instance-invoking process in Grid, two additional messages (messages 3 and 4) are brought by the session authority. Note that message 1 needs to be protected by additional security measures. Because the new resource has not been generated, and thus the MAC associated with the message cannot be calculated. After the resource has been generated, the integrity and freshness of messages 2 ~ 6 can be protected by using MACs and nonces.

We have implemented two experimental systems. The first experimental system (*ES*1 for short) consists of a SA service and three experimental services. In this experiment, three experimental services repeatedly invoke each other in the sequence demonstrated in Figure 2 until a particular amount of service instances have been spawned and introduced to the SA.

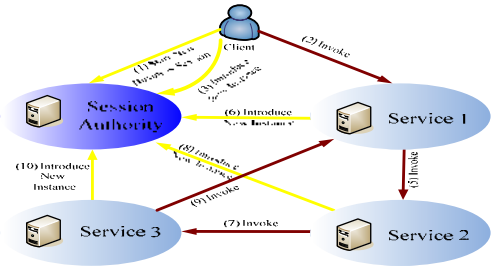


Figure 2: Experimental system with the SA

Compared with *ES*1, the system structure of the second experimental system (*ES*2 for short) is relatively simple. *ES*2 consists of three experimental services. In the experiments, the experimental services invoke each other repeatedly until a particular amount of service instance has been generated. *ES*1 is used to evaluate the performance of the system incorporated with the SA, whilst *ES*2 is used to evaluate the performance of the Grid infrastructure. By comparing the experimental results obtained from *ES*1 and *ES*2, we can evaluate the influence introduced by our authentication mechanism on the performance and scalability of Web service systems.

3.1 Experiments on a Single Computer

We deploy the experimental systems on a single computer for two reasons. Firstly, in order to precisely evaluate the overhead introduced by our authentication security protocols (e.g., generating key pairs, generate MACs for messages, etc.) we need to remove the influence brought by the time consumed on transporting messages. Secondly, if an experimental system is deployed on different computers, operations in the system may be executed in a concurrent fashion. Deploying the experimental system can help us to evaluate the performance of the systems in the worst case where all the operations of the system are executed sequentially. In this sub-section all the experiments are deployed on a single computer unless mentioned otherwise.

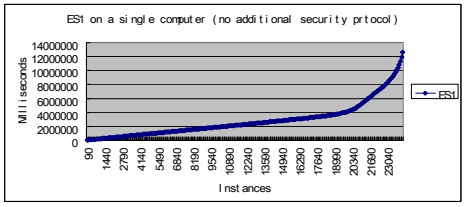


Figure 3: *ES*1 deployed on GT4

The experiment illustrated in Figure 3 is used to evaluate the scalability of *ES*1. In this experiment, more than 24,000 instances are generated and introduced to the SA. From the beginning of the experiment, the experimental system stays in a stable state. The time consumption of the system is proportional to the number of the generated instances, until over 16,000 instances are generated. After that, the performance of the experiment becomes bad, and the system finally stops due to the lack of the memory. In this experiment, no additional security protocol is used to secure the messages transported between instances. In experiment shown in Figures 4 and 5, we incorporate *ES*1 and *ES*2 with messages level protocols (i.e., secure conversation and secure message provided by GT4).

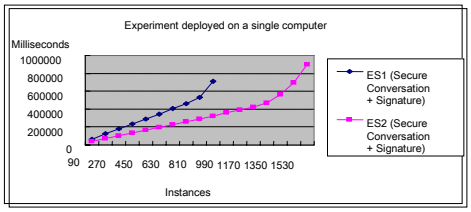


Figure 4: *ES*1 and *ES*2 on GT4 + Secure Conversation

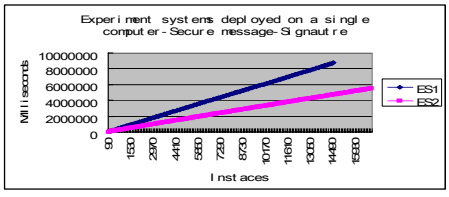


Figure 5: *ES*1 and *ES*2 on GT4 + Secure Message

When *ES*1 is incorporated with the secure conversation protocol, that is, the secure conversation protocol is used to generate signatures for every message transported in the experiment, the time consumption of the experimental system starts increasing non-linearly after over 720 instances are generated. The same phenomenon occurs in *ES*2 after 1040 instances are generated when *ES*2 is incorporated with the secure conversation protocol. As illustrated in Figures 4 and 5, the time consumption of *ES*1 is about as twice as that of *ES*2, while *ES*1 can realize fine-grained authentication at the instance level.

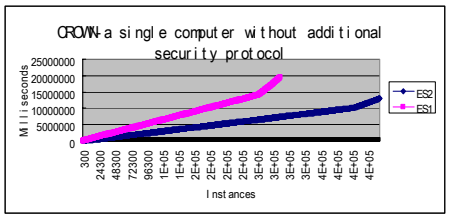


Figure 6: Compare between *ES*1 and *ES*2 on CROWM

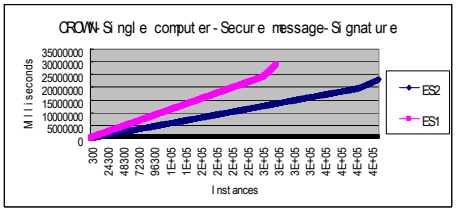


Figure 7: Compare between *ES*1 and *ES*2 on CROWN + Secure Message

We have successfully implemented our design in the large-scale CROWN (China Research and Development environment Over Wide-area Network) Grid. CROWN aims to promote the utilization of valuable resources and cooperation of researchers nationwide and world-wide. In the experiments presented in Figure 6, our experiments system executed without any additional security protocol. In the experiments presented in Figure 7, our experimental system uses the security message protocol to protect the integrity of the messages transported between the instances generated in the experiments. In these experiments, our authentication system can stay in a stable state until over 260,000 instances are generated. This demonstrates that although the authentication protocol introduces some overheads to the system, the scalability of the authentication system is still very good.

3.2 Distributed Deployed Experiments

After evaluating the performance of the experimental systems deployed on a single computer, we deploy the experimental systems in a distributed way and further evaluate them in a more realistic environment. In the experiments introduced in this sub-section, the experimental systems are distributed unless mentioned otherwise. The scalability of the experimental systems deployed in a distributed way is much better than on a single computer. As illustrated in Figure 8, the time consumption of the distributed *ES*1 increases linearly, until more than 70,000 new instances are generated. When incorporated with the secure conversation protocol (see Figure 9), *ES*1 executes stably until over 3,000 instances have been generated. In the experiments shown in Figures 10 and 11, the time consumption of *ES*1 is proportional to the number of the new generated instances until over 300,000 instances are generated.

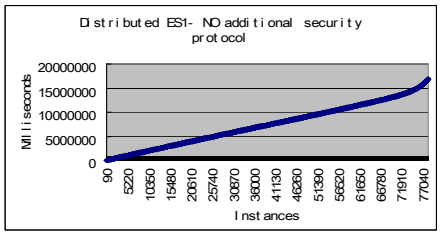


Figure 8: *ES*1 deployed on GT4

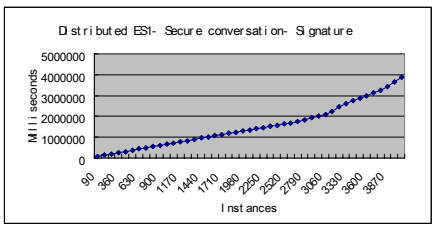


Figure 9: *ES*1 deployed on GT4+ Secure Conversation

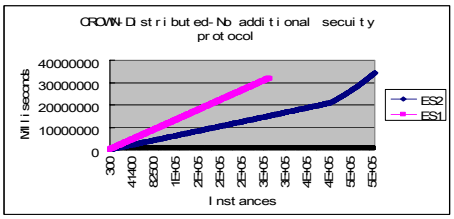


Figure 10: Compare between *ES*1 and *ES*2 on CROWM

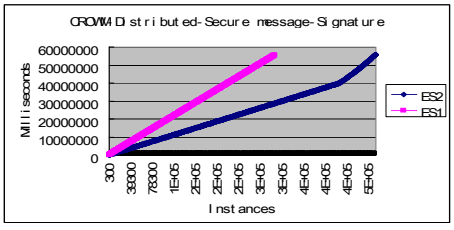


Figure 11: Compare between *ES*1 and *ES*2 deployed on CROWN + Secure Message

3.3 Analysis

The experimental results presented above demonstrate that our authentication system can realize the instance level authentication with a reasonable cost. Additionally, we conclude that the scalability of our experimental systems is influenced by several factors:

- The security protocols which our authentication system is incorporated with. When incorporated with different security protocols, the scalability of *ES*1 notably varies. For example, when *ES*1 is deployed on a single computer and incorporated with the secure conversation protocol, the time consumption increases nonlinearly after over 720 instances are generated. When *ES*1 is cooperated with the secure message protocol, the system can execute stably even after generating more than 13,000 instances.
- The amount of the memory. In *ES*1, the SA stores the information about the session partners in memory, and the resources of experimental services are stored in memory as well. Thus, the amount of the memory is critical to the scalability of *ES*1. When the experimental systems are deployed on a single computer, the SA and the experimental services have to share the limited memory. This is a main reason that the scalability of the distributed experimental systems is much better than the experimental systems deployed on a single computer.
- The time and memory consumed on reading and writing the log files. In our experiments, all the log information is stored in a log file. As the length of the log file increases, the time and memory consumed on reading and writing the log file increase too.

4. An Instance-Level Authorization Mechanism

Beside instance-level authentication, instance level authorization is another essential requirement to generate secure service-based business processes [8]. Previous research in authorization for business processes rarely considers the scenarios where multiple parties cooperate in a peer-to-peer way, and a lot of issues need to be further explored.

4.1 Limitations of the Current Authorization System in Workflows

Workflow is a popular technology to implement automatic business processes. Despite some research [2, 4] has been done to address the instance level authorization and authorization issues in workflow systems, existing approaches can not be adapted to flexible business processes that consist of a large number of instances which are dynamically bound and collaborated. The workflows considered in these approaches are normally compliant with the workflow reference model [6] proposed by the Workflow Management Coalition. In this model business specifications are executed by a centralized workflow engine, and the participants of workflows are passive. The participants just collect work items from their work lists, achieve jobs, and send results back. All the interactions among instances are managed by the workflow engine. Thus, the access control rules are configured by the administrator statically, and the participant instances have no privilege to dynamically specify their own access control policy. In contrast with the business process supported by the conventional workflow systems, Web service business processes rely on the peer-to-peer interactions between participant instances. This is because Web services are designed for the cross-organizational environments, and have to face the lack of a central location for the middleware [1]. Therefore, the communication between service instances can be very complex, and it is common that multiple instances cooperate to achieve a business goal. For example, in a business session, a session partner invokes a medical data processing service, which can be accessed openly, and generates an instance to processing the session partner's medical data, which, however, only can be accessed by special instances involved in this business session, such as a hospital, thus the invoker needs to assign its own access control policy to this new invoked instance. In order to address this issue, we propose an instance-level authorization mechanism which is designed for Web service business processes. Compared with other authorization solutions for business processes, our authorization system has the following features:

- The invoker can dynamically specify the access control policies of the invoked instance at run-time, and thus the access

control relationship between instances is more flexible.

- With the assistance of the instance-level authentication system, our authorization mechanism can define the business session constraints within the access control policies.

4.2 Design of Authorization Mechanism

As Web services are loosely coupled and can be bound at run-time, the number of the session partners involved within a business session may keep changing. Therefore, it is difficult for an instance to obtain real-time knowledge about its session partners when it is involved in a peer-to-peer business session. For instance, when an invoker sets the access control policies to a new invoked instance, it may not know the identifiers of the session partners who are going to access this new instance. On extreme occasions, the instances supposed to access the new instance have not been spawned yet. Therefore, identity-based authorization systems are not suitable, and we propose an attribute-based authorization mechanism instead.

We adopt XACML (eXtensible Access Control Markup Language) to express fine-grained access control policy. As illustrated in Figure 12, our authorization system consists of two components, the authorization service (AuthzService for short) and the SA service.

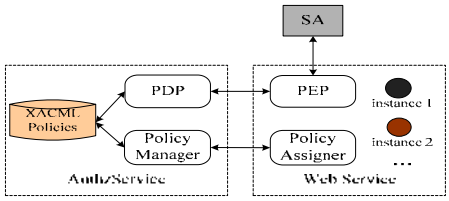


Figure 12: The architecture of the system

The authorizing processes can be broken down to two parts: assigning the access control policies to the new created instance and making authorization decisions with respect to the requests to access instances. During the process of policy assigning, the instance invoker can contact the *Policy Assigner* of the target service to specify access control policies, and then the *Policy Assigner* will call the *Policy Manger* in the AuthzService to write this policy and make it available to the *PDP* (Policy Decision Point). When there is a request for this instance controlled by dedicated policy, The *PEP* (Policy Enforcement Point) of this instance will firstly contact the SA and query the authentication result. Secondly the *PEP* accesses the *PDP* of the AuthzServcie; the *PDP* then makes an authorization decision according to this authenticated Token. In practice, the AuthzService can be implemented in various ways. For example, the AuthzService can be implemented as a centralized service which deals with the requests from all the session partners within the business session. The AuthzServices also can be implemented in a decentralized way; that is, each AuthzService only manages the service instances within a local domain, and multiple AuthzServices are associated with a business session.

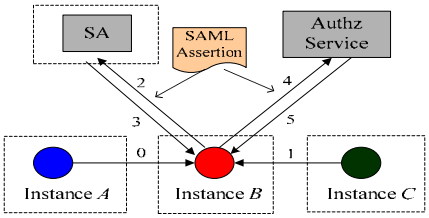


Figure 13: An example of the authorization process

Figure 13 presents an example of the authorization enforcement. In this example there are three instances in a business session, *A*, *B* and *C*. *A* invokes *B*, and specifies an access control policy encoded with XACML for the new generated *B*. *C* also intends to access *B*. Assume that *A* has stored the policy into the authorization service of *B* (step 0 in Figure 13). The steps of the authorization enforcement in Figure 13 are presented as follows:

1. $C \rightarrow B:\{request, id_C, attrs_C\}$. *C* sends the *request* of service access, its instance identifier id_C, and other related attributes information $attrs_C$ to *B*.
2. $B \rightarrow SA:\{auth_{request}\}$. *B* encapsulates *C*'s information into a SAML assertion $auth_{request}$ and sends $auth_{request}$ to the SA to query whether *C* is in the same business session with *B*.
3. $SA \rightarrow B:\{auth_{result}\}$. The SA verifies whether *C* is involved in this business session, and replies a SAML assertion $auth_{result}$ which consists of the authentication result to *B*.
4. $B \rightarrow AuthzService:\{auth_{result}, id_B\}$. After receiving the response from the SA, *B* encapsulates the authentication result and its instance identifier to the AuthzService.
5. $AuthzService \rightarrow B:\{authz_{result}\}$. The AuthzService makes an authorization decision according to the XACML polices

that *B* assigned in advance and returns the authorization result back.

During the processes of authentication and authorization, SAML (Security Assertion Markup Language) is used to describe related security assertions in the messages. For example, the attributes (e.g., identifier) of an instance ought to be encapsulated according to the SAML specification.

5. Conclusion

Conventionally, business process supporting techniques (e.g., workflow) mainly deal with the issues of implementing business process within an organization, and business processes supported by these techniques are normally implemented in the client/server model. A Web service business process may have to execute in a cross-organizational environment where the client/serer model is not suitable. Web services have to achieve business objectives through peer-to-peer cooperative interactions [1, 3]. The structures of Web service business processes thus can be more flexible and complex than conventional business processes. Because session partners cooperate in a peer-to-peer way, a session partner may lack the real-time information about its session partners. All these issues bring new challenges to security systems; a lot of security issues also need to be carefully considered. In this paper, we present our efforts in instance-level authentication and authorization. An instance-level authentication system is proposed. Compared with traditional authentication systems, this system is able to generate the identities for session partners and authenticate them at run-time. Our instance-level authentication system can help a service instance to distinguish its session partners from the instances in other sessions, and thus generate a security boundary for business sessions. However, considering the potential competitions among session partners and a lot of other security requirements (e.g., the separation of duty), we propose an instance-level authorization system so as to further improve the security of business sessions. In most conventional authorization systems for business processes, the access control relation within a business session is statically predefined by the administrator of the business session. In Web service business sessions, session partners works in a peer-to-peers way and their relationship is more flexible and difficult to be precisely predefined. Therefore, our authorization system enables session partners to dynamically assign access control policies for the new invoked instances and ensure the correctness and the security of the cooperation between the session partners.

We deployed our experimental systems on the GT4 and CROWN Grid infrastructures. These experimental systems are mainly used to evaluate the scalability and the performance of our instance-level authentication system. In some experiments, our authentication system can execute stably until over 260,000 instances are generated. Such experimental results demonstrate the scalability of the instance-level authentication system is good. Additionally, compared with the performance of service-level security mechanisms, the overhead introduced by our instance-level authentication system is reasonable. We are now in the process of evaluating the performance of the instance-level authorization system and going to present the results in the papers published in the future.

Reference

[1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, "Web Services," *Springer Verlag*, 2003.

[2] V. Atluri, WK Huang, "An Authorization Model for Workflows," *Proc. the 5th European Symposium on Research in Computer Security*, Lecture Notes in Computer Science, Vol. 1146, Springer-Verlag, pp. 44-64, 1996.

[3] G. B. Chafle, S. Chandra, V. Mann and M. G. Nanda, "Decentralized Orchestration of Composite Web Services," *Proc. the 13th international World Wide Web conference on Alternate track paper & posters*, pp. 134-143, May 2004.

[4] D. Domingos, A. Rito-Silva, P. Veiga, "Authorization and Access Control in Adaptive Workflows," *Proc. 8th European Symposium on Research in Computer Security*, pp. 23-38, 2003.

[5] S. Hada and H. Maruyama, "Session Authentication Protocol for Web Services," *Proc. 2002 Symposium on Application and the Internet*, pp. 158-165, Jan. 2002.

[6] D. Hollingsworth, "Workflow Management Coalition: The Workflow Reference Model," *Technical Report WFMC-TC-1003*, Workflow Management Coalition, Brussels, Belgium, 1994.

[7] T. Y. C. Woo and S. S. Lam. "A Semantic Model for Authentication Protocols," *Proc. IEEE Symposium on Research in Security and Privacy*, pp. 178--194, Oakland, California, May 1993.

[8] D. Zhang and Jie Xu, "Securing Instance-Level Interactions in Web Services," *Proc. 2005 ISADS IEEE*, pp. 443-450, April 2005.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

A user-friendly approach to computational grid security

B. Beckles

University of Cambridge Computing Service, New Museums Site, Pembroke Street, Cambridge CB2 3QH

P. V. Coveney

Centre for Computational Science, Department of Chemistry, University College London, Christopher Ingold Laboratories, 20 Gordon Street, London WC1H 0AJ

P. Y. A. Ryan

School of Computing Science, University of Newcastle, Newcastle upon Tyne NE1 7RU

A. E. Abdallah

Institute for Computing Research, Faculty of Business, Computing, and Information Management, London South Bank University, 103 Borough Road, London SE1 0AA

S. M. Pickles, J. M. Brooke, and M. McKeown

Manchester Computing, Kilburn Building, The University of Manchester, Oxford Road, Manchester M13 9PL

Abstract

Many of the existing security components and frameworks for computational grid environments either suffer from significant usability issues for end-users and/or administrators, or their administration and deployment is extremely complex and resource-intensive. This has lead to a situation where using such environments securely is so difficult that end-users either refuse to use them or else deliberately use them in an insecure fashion. In this paper we describe work underway to provide more user-friendly security mechanisms for computational grid environments.

1 Introduction

Although computational grid environments are being increasingly considered for use for scientific computing, there is still disagreement about the precise definition of "computational grid" and related terms ([1]). For our purposes, we define "computational grid environment" to be a distributed computing environment which is transparent across multiple administrative domains (following [2]). While the apparent benefits of working in such environments have been widely publicised (e.g. [3]), it is still the case that there is relatively little use of "grid technology" for serious computing projects ([4], [5]). In particular, routine use of computational grid environments by the academic community is still far from widespread, with many researchers being reluctant to use the existing environments (as noted in [6], [7]).

One class of problem faced by end-users of these computational grid environments is to do with the usability of the security mechanisms/frameworks usually deployed in these environments ([8]). This is particularly a problem regarding authentication in these environments ([8]). In fact, reports from end-users have revealed that some users will knowingly use the environments in a manner they know to be insecure because using it in the mandated "secure" fashion is too difficult ([8]). Some potential users have even refused to use these environments at all if they are forced to use the "usual" authentication mechanism (digital certificates) used by the existing environments ([9]). Thus the usability problems of the existing grid security solutions are so serious as to be a major barrier to the adoption of grid technology by the wider community.

In this paper we outline some of the major usability and other related problems with the existing security solutions in use in computational grid environments and then discuss our proposed methods of addressing these problems. The key aspect of our development methodology is the adoption of a "user-friendly" approach to grid security, which combines aspects of usability engineering with formal methods in security engineering. This ensures that not only are our security components easy for our end-users to use, but also that they are theoretically sound.

2 Grid security: the problems

Before discussing the problems we have encountered with the existing grid security solutions we define a few terms that we shall use frequently in our discussion of these problems. We define "usability" to be the extent to which the user (subjectively) feels that the software has

successfully fulfilled their requirements for that software. Depending on the context, the user in question may be the end-user of the software, the system administrator installing, maintaining or administering the software, etc. We consider software to be "user-friendly" if its usability is high and if the user is able to use it appropriately in such a manner that they are either unaware that they are using it or that its impact on their user experience is positive.

In the context of computational grid infrastructure, we use "heavyweight" to mean software that is some combination of the following:

- complex to understand, configure or use;
- has a system "footprint" (disk space used, system resources used, etc.) that is disproportionately large when considered as a function of the frequency and extent to which the user uses the software;
- has extensive dependencies, particularly where those dependencies are unlikely to already be satisfied in the operating environment, or where the dependencies are themselves "heavyweight";
- difficult or resource-intensive to install or deploy;
- difficult or resource-intensive to administer; and/or
- not very scalable.

We use the term "lightweight" to refer to software that is not "heavyweight", i.e. it possesses none of the features listed above, or only possesses a very small number of them to a very limited extent. Thus lightweight software would be characterised by being:

- not too complex for end-users, developers and administrators to use and understand in their normal work context,
- easy to deploy in a scalable manner, and
- easy to administer and maintain.

See [6] and [10] for a fuller discussion of lightweight and heavyweight grid middleware.

The problems we have encountered with the current grid security mechanisms and frameworks tend to fall into two main categories. There are those problems that seem to be common to most of the existing grid security solutions, e.g. excessive complexity, poor scalability, high deployment and maintenance overheads, etc. We observe that all these solutions have a common characteristic that we believe is responsible for these problems, namely they are all very heavyweight solutions to the security problems they address.

Then there are those problems that are specific to the particular aspect(s) of security (e.g. authentication, authorisation, etc) with which the solution is concerned, e.g. credential management. First we shall outline those problems that we have encountered that we believe are due to the heavyweight nature of the existing solutions, and then we shall discuss problems specific to particular aspects of computational grid security.

2.1 Heavyweight security solutions

Many of the problems inherent in heavyweight grid middleware have been discussed in detail elsewhere ([6], [2], [10]), and so we shall not discuss them here. Instead we outline the specific problems that the heavyweight nature of the existing grid security solutions causes in the context of the security of computational grid environments. (It is important to note, however, that, security considerations aside, heavyweight grid middleware is, by its very nature, a significant barrier to the wider uptake of grid technology ([6], [11], [2]).) The specific problems that such grid middleware causes in a security context include the following:

- *Complexity:*

Amongst security professionals it is well known that complexity is the enemy of security, indeed, some security professionals have described complexity as the *worst* enemy of security ([12]). The existing grid security solutions are extremely complex, and this makes them very difficult to configure or use, and, most crucially, to understand ([13], [8]). Thus it is very difficult even for experienced system administrators to correctly install and configure these security solutions ([14]).

- *Extensive dependencies:*

The security of a piece of software is – at a minimum – dependent on the security of its components and dependencies. It is also dependent on many other factors, but if the security of any of its components and dependencies is breached then the likelihood is high that the security of the software as a whole will also be violated (although there are techniques, such as privilege separation ([15]), which can help mitigate this). The software's components and dependencies thus form a "chain of trust", which is only as strong as its weakest link ([12]). It therefore follows that, all else being equal, software with extensive dependencies is both more likely to be vulnerable to security exploits, and also more difficult to secure in the first place, than software with few dependencies.

It is also worth mentioning in this context that software which installs its own versions of system libraries (or is statically compiled with respect to those system libraries) is also likely to be more insecure. This is because its versions of these

libraries are unlikely to be as readily upgraded in response to security alerts as the system libraries, in part because system administrators may well not be aware that the software uses its own versions of these libraries.

- *Scalability:*

Many of the current grid security solutions are not particularly scalable ([16], [17]). Given that, at least in principle, computational grids can consist of thousands of individual users and machines and hundreds or thousands of administrative domains, it is clear that grid middleware whose scalability is poor will not be suitable for medium to large grids. One aspect of this lack of scalability that has serious security implications is the requirement for the security configuration of each node and/or client in a grid environment to be individually configured ([8], [14]). As one might expect, this leads to some nodes/clients being incorrectly configured, and in an environment with a large number of nodes this may remain unnoticed for a considerable period of time.

In addition, where this configuration must be maintained on a regular basis – particularly if the software is designed to stop working if its configuration is not kept up-to-date – it is not unusual for users to have found ways of circumventing this. For example, where up-to-date certificate revocation lists (CRLs) are required for successful operation, often no CRLs will be installed to avoid the software failing when the CRL is no longer current ([8]).

- *Usability:*

From the definition of "heavyweight software" given above it should be clear that it is inherently difficult for such software to have high usability or to be user-friendly ([10]). In addition, an examination of the software development process for most of the existing grid security solutions reveals very little evidence of usability considerations playing a significant role in their design. Given the wealth of usability-related problems ([8]) with the Grid Security Infrastructure (GSI) ([18]) – the principal authentication mechanism used in existing grid environments – it seems unlikely that its design was informed by many usability concerns. The authors are only aware of two grid security components/solutions in active use – MyProxy ([19]) and PERMIS ([20]) – where usability concerns have been explicitly considered at some point in their development/deployment ([19], [21]), and even with those products it is not clear whether usability was a core concern of their designers.

Unfortunately, since security mechanisms whose usability is poor in the environment in which they are deployed are inherently insecure ([22], [23]), it is therefore likely that most of the existing grid security solutions will be deployed or used in an insecure manner in practice. Further, usability is not a "feature" that can be bolted on at the end of the development process; usable systems are only possible if usability is a central concern of the designers at all stages of the system's design and development ([24], [25], [26]). It is thus unfortunately the case that most of the existing grid security solutions are unlikely to be usable or secure in practice.

2.2 Authentication

As mentioned above, the principal authentication mechanism in current computational grid environments is GSI, and there are a number of serious problems with this mechanism; these are discussed in some detail in [27], [8] and [14]. Perhaps the most significant point that has emerged from these examinations of the problems with GSI is that the digital certificates upon which it relies are "cognitively difficult objects" for its users ([8]). This is borne out by the fact that some potential users of grid computing environments have refused to use those environments if they required a digital certificate in order to do so ([9]). It would thus seem probable that any better authentication solution for grid environments must not rely on users making active use of digital certificates ([8]). Ideally, users should not have to have such certificates at all ([28], [8], [14]).

2.3 Authorisation

The current computational grid environments provide very limited authorisation mechanisms ([16], [17]); most of them rely on simple "allow" lists in manually maintained text files. Although more sophisticated mechanisms are available (e.g., CAS ([29]), VOMS ([30]), PERMIS ([20]), Shibboleth ([31])), few of these are currently in active use in the existing environments. Some of these mechanisms (e.g. CAS, VOMS) are based on GSI and thus inherit its usability and security problems. In addition, they are all, to a greater or lesser degree, heavyweight (in the sense defined above). This leaves system administrators with two choices, neither of which is satisfactory: they can either rely on an overly simple mechanism that does not scale well ("allow" lists in text files), or they can attempt to use one of the more sophisticated, heavyweight, solutions, with all of the associated problems inherent in such solutions.

2.4 Auditing

The auditing features of current computational grid environments are, at best, rudimentary. GSI, the

principal authentication mechanism, relies wholly on the integrity of users' credentials being maintained – the only auditing information provided is the distinguished name (DN) of the digital certificate and the IP address from which the authentication request appears to originate. As discussed in [8], the likelihood of the integrity of the digital certificate being compromised is high. Also, in the general case, an entity's apparent IP addresses cannot be relied upon to belong to it (due to IP address "spoofing"). Thus, the audit information provided is often of little value when attempting to detect or clean up after a security incident.

Very little work has currently been done in the area of the auditing requirements of computational grid environments, and so it is not clear what information needs to be collected, or how it should be stored, processed, etc ([32]). However, it is clear that most of the existing security solutions do not address these problems in any detail. Indeed, many of them (e.g. GSI), do not follow accepted best practice of storing the audit data at a location remote to where that data was gathered. This means that, in the event of the security of a machine in the grid environment being breached, the attacker will be able to modify the audit data that might otherwise reveal how they obtained access to the machine.

3 Solution Requirements

From the discussion above, and other work in the areas of grid middleware design ([6], [10], [9]), grid security ([27], [28], [8], [14]), and usable security ([22], [23], [33], [34], [21]), we abstract some very general requirements that any grid security solution should possess. We then consider some of the specific security requirements in the areas of authentication, authorisation and auditing.

3.1 General requirements

As should be clear from the discussion in Section 2.1, heavyweight grid security solutions are unlikely to improve the current situation. Indeed, the seriousness of the problems inherent in heavyweight grid middleware is arguably one of the most significant factors in the relatively low use of current computational grid environments ([7], [10]). Thus a core requirement for any grid security solution is that it is *lightweight* (in the sense defined above).

As discussed above, security solutions whose usability is poor are inherently insecure, and usable security solutions require usability to be a central concern of the developers at all stages of the solution's design and development. Thus grid security solutions must have *high usability* in their deployed environments, and usability must be a core concern of their designers. As discussed in [10], this requires *continuous user involvement* in the design and development processes from their earliest stages.

Given the large investment (in terms of resources allocated) in the existing computational grid environments, it is clear that, despite their relatively low usability and the high likelihood of their security being violated, they cannot simply be abandoned. Thus our solution also needs to be *interoperable* with the existing computational grid environments.

As security professional Bruce Schneier notes in [12], "In cryptography, security comes from following the crowd", and this is true of secure software in general. From a security standpoint it is generally undesirable to re-invent security primitives (especially cryptographic primitives), since these will, of necessity, have received less testing and formal security verification than existing primitives that are widely used. We therefore intend to *use appropriate existing cryptographic techniques* rather than developing any new ones. In addition, we intend to *use existing cryptographic toolkits and security components*, where we can do so without negatively impacting on our other design goals (such as usability).

In summary, our solution must be:

(a) lightweight, i.e.
   - of low complexity,
   - easy to deploy,
   - easy to administer,
   - minimal in its external dependencies, and
   - scalable.

(b) highly usable by its intended users,

(c) developed in conjunction with its intended users,

(d) interoperable with the existing computational grid environments (or at least with the most common ones),

(e) implemented using existing cryptographic techniques, and

(f) maximal in its use of appropriate existing cryptographic toolkits and other security components.

See [22], [33], [35], [6], [10] and [21] for more complete discussions on appropriate design and development methodologies and guidelines for meeting requirements (a) to (c).

3.2 Authentication

As discussed in Section 2.2, it is vital that *end-user interaction with digital certificates be minimised*, ideally to the point where end-users have no interaction with digital certificates. A number of

methods have been proposed for doing this ([28], [8], [14]). We have chosen to adopt the method described in [28] as it is designed to be interoperable with the existing grid authentication mechanisms, highly scalable, extensible and represents a significant improvement in its auditing facilities over the existing mechanisms. Of equal importance, its development methodology fully embraces the need for usability to be a core concern of any security solution and for user involvement in the design and development processes.

3.3 Authorisation

It seems intuitively obvious that a truly general authorisation framework that would be applicable to any possible scenario in a computational grid environment is likely to be extremely complex and its implementations would quite probably be very heavyweight. The general nature of authorisation frameworks such as PERMIS and Shibboleth is probably precisely what causes them to be such heavyweight security solutions and so unsuitable for our purposes. As discussed in [10], it is probably impossible to develop usable grid middleware that can be used in all the problem domains that might be of interest. Instead, smaller domain-specific middleware is needed.

Thus we do not attempt to re-invent a general authorisation framework. Instead, we propose to concentrate on some specific problem areas of interest and develop a *lightweight* authorisation framework appropriate for those areas. A design goal for this framework is that it is *extensible*, so that others working in similar problem areas are likely to be able to make use of it. As noted in Section 3.1, *interoperability* is a crucial requirement, and so our initial investigation will be into wrapping the flat file authorisation tables used in most existing computational grid environments, in a manner analogous to that proposed in [28] for the existing authentication mechanisms.

3.4 Auditing

As discussed in Section 2.4, the auditing requirements for computational grid environments are not yet well understood, and the existing facilities are quite basic. Further research is necessary before the requirements in this area can be definitively stated. However, it seems likely that *conforming with accepted best practice for network auditing* (e.g. storing audit data at a location remote from where it is collected, not relying on easily falsifiable identifiers such as IP addresses, etc.) would be a sensible place to start. The mechanism proposed in [28] attempts to improve the reliability of the data collected for audit purposes and we propose to use this as a starting point for further development.

4 Development Methodology

Our development methodology is a combination of user-centred security methodologies (such as [22], [35]), drawing heavily on the user-centred design methodologies (e.g. [25], [36], [37]), in conjunction with formal methods drawn from security analysis (e.g. [38], [39]). There are two principal strands to our development strategy: the development and security verification of working software solutions, and research into, and formal analysis and modelling of, the security space (computational grid environments) in which that software is to be deployed.

Both usability and security requirements are situated requirements, and, as such, are crucially dependent on context. As mentioned above (and discussed in more detail in [6] and [10]), the attempts to produce truly general solutions has led to heavyweight grid middleware whose usability is low and this has directly contributed to the low adoption of grid technology by the academic community. To avoid repeating these mistakes, we are situating our project in a specific context, namely the RealityGrid Project ([40]) and its chosen middleware, WEDS ([2]) and the AHE ([7]). A design goal of our software is thus tight integration with the middleware used by the RealityGrid Project.

As discussed in [41] and [10], a prerequisite for effective development of any software solution is a detailed knowledge of the requirements of the stakeholders for whom that solution is being developed. Thus the first stage in our software development plan is a requirements capture exercise using techniques from the user-centred design methodologies. The results of this exercise are fed into both the formal modelling and analysis of the security problem space under consideration, and the software design process. Low fidelity prototyping (e.g. [42]) is used to ensure that the gathered requirements accurately reflect the users' needs. Our development cycle thus looks something like this:

1) Requirements capture and analysis;
2) Formal modelling and analysis;
3) Software design;
4) Formal verification of security of design;
5) Usability testing using low fidelity prototypes.

The above cycle is iterative (note that the different phases may partially overlap) with each phase dependent on the successful completion of the previous phase. If a phase cannot be completed successfully, earlier phases are repeated

with the current design and suggested changes from the failed phase as input. This cycle iterates until a stable design is produced.

Note that no program code (even as a prototype) is written until a stable design has been produced, and this design has passed usability trials and security verification. This helps to ensure that software development is user-driven rather than programmer-driven, and that the software design is responsive to user concerns. After the stable design is produced, a software prototype of this design is then developed, which is integrated with our target middleware in the following cycle:

1) Development of software prototype;
2) Integration with target middleware;
3) Usability testing;
4) Formal verification of security of integrated prototype.

The above cycle is iterative in a similar manner to the previous cycle. This cycle iterates until a working software prototype has been integrated with the target middleware and passed its usability trials and security verification.

We have adopted a component based approach to the authentication and authorisation mechanisms we are trying to develop, so these are developed as separate components, the development of each following the development cycle described above.

5 Security Improvements

In this section we discuss how our proposed solutions and our software development methodology will improve the security of computational grid environments. First we discuss the impact of our development methodology on the security of our solutions, and then we discuss how our solutions address some of the security issues present in the current computational grid environments.

5.1 Development Methodology

Our development methodology incorporates software development principles and practice (such as [22], [33], [35], [34]) proven to increase the security of systems by addressing usability concerns. In addition, we undertake a continuous programme of usability testing to ensure that our usability goals have actually been achieved. Thus not only will our software have been designed in accordance with proven usability design principles, but we will have tested the software in usability trials to ensure that it genuinely possesses a high degree of usability for its intended user community. Together, these measures guarantee that – at least in usability terms – our solutions should be major improvements over the current grid security solutions.

In addition, our development methodology incorporates formal security analysis and security verification of both our software design and its implementation. By using formal security methods we can ensure that our software's security model is at least theoretically sound. Whilst this does not, of course, guarantee that the software will be used securely in practice, if any software deployment is to be secure, a prerequisite is that its security model is theoretically sound. We are unaware of any formal security analysis or verification of the existing grid security solutions. Once the final implementation of our software has undergone formal security analysis and verification, we believe it would be reasonable to have greater confidence in our software being secure than in the existing solutions.

5.2 Authentication

From the discussion above (and see [8] and [14] for further details) it is apparent that, in current computational grid environments, there is a significant risk of users' credentials being obtained by unauthorised individuals, and this is principally due to usability issues with the current security mechanisms. Obtaining user credentials would allow attackers to successfully exploit these environments *without* needing to discover and exploit security vulnerabilities in the program code of the security mechanisms. We intend to address the usability issues with the current mechanisms (see Section 3.2), and, as discussed in Section 5.1, will undertake extensive usability testing to *ensure* we have addressed these issues. This means that our software should significantly reduce – or possibly even eliminate – the risk of attackers obtaining users' credentials through the usability deficiencies of the environment's security mechanisms.

5.3 Authorisation

Environments that currently use the existing heavyweight authentication solutions are likely to have inappropriate authorisation policies in place, i.e. the complexity of the solution is such that the actual authorisation policy implemented may not match the intended authorisation policy in the administrator's mind. If the implemented policy is too lax, then unauthorised individuals may be able to access the environment, or users may be able to perform unauthorised actions. If the implemented policy is too strict, this is likely to be quickly discovered by users, who will complain about being unable to perform legitimate actions. This will lead to a relaxation of the policy, and may

well lead to the policy being relaxed to an inappropriately high degree. This may happen because of the dissonance between the administrator's understanding of the authorisation mechanisms and the actual workings of these mechanisms (which led to the mistaken policy being implemented in the first place). Alternatively, fear of further angering the user community may lead the administrator to err on the side of "caution", i.e. to implement an extremely permissive policy so as to be certain that the users are able to do what they should be allowed to do.

As described in Section 5.1, our development methodology actively addresses the usability issues that cause such dissonance. Consequently, our authorisation component will represent significant improvements for administrators (and so for users) over the current situation, i.e. it will be much more likely that administrators will be able to correctly set appropriate authorisation policies.

5.4 Auditing

As discussed in Section 2.4, the auditing facilities of the existing computational grid environments are not very sophisticated, and the auditing requirements are not very well understood. This means that it is likely to be difficult to diagnose attacks against these environments. We intend to actively investigate this area as a better understanding would allow us to implement better auditing facilities. However, as discussed in Section 3.4, merely by following acknowledged best practice for network auditing we will have improved the current situation significantly.

6 Preliminary Work

As described in [28] and [8], our proposed replacement authentication mechanism is nearing the end of its design phase – see [8] for details of usability concerns addressed in the design process. We anticipate soon having some software prototypes for integration with our target lightweight grid middleware ([2], [7]). Once this is done we will begin usability testing of these prototypes. As further development of the other security mechanisms proceeds, we will gradually incorporate prototypes of these mechanisms in accordance with the software development methodology outlined in Section 4.

7 Conclusion

The existing grid security solutions suffer from a number of deficiencies that are serious barriers to wider adoption of grid technologies by the academic community. Many of these deficiencies stem from the heavyweight nature of the solutions in question, and from their low usability. The low usability of these solutions is probably a consequence of the fact that usability considerations were not central to their design.

In this paper we have described how our "user-friendly" approach to grid security seeks to mitigate the existing deficiencies by closely involving the user in the design and development processes. By combining this with formal methods in computer security we aim to produce lightweight security solutions that are easy to use for our target user population and which are theoretically sound.

Acknowledgements

This work is supported by EPSRC through the *User-Friendly Authentication and Authorisation for Grid Environments* project (EP/D051754/1).

References

[1] Hurley, J. "THE GRID IDENTITY CRISIS: DEFINING IT AND ITS REALITY", GRIDtoday, Vol. 1, No. 10 (19 August, 2002): http://www.gridtoday.com/02/0819/100248.html

[2] Coveney, P., Vicary, J., Chin, J. and Harvey, M. Introducing WEDS: a WSRF-based Environment for Distributed Simulation (2004). UK e-Science Technical Report UKeS-2004-07: http://www.nesc.ac.uk/technical_papers/UKeS-2004-07.pdf

[3] Evans, B. "Grid Computing: Too Big To Be Ignored". *InformationWeek*, 10 November 2003: http://www.informationweek.com/story/showArticle.jhtml?articleID=16000717

[4] McBride, S. and Gedda, R."Grid computing uptake slow in search for relevance". *Computerworld Today*, 12 November 2004: http://www.computerworld.com.au/index.php?id=138181333

[5] Ricadela, A. "Slow Going On The Global Grid". *InformationWeek*, 25 February 2005: http://www.informationweek.com/story/showArticle.jhtml?articleID=60402106

[6] Chin, J. and Coveney, P.V. Towards tractable toolkits for the Grid: a plea for lightweight, usable middleware. (2004). UK e-Science Technical Report UKeS-2004-01: http://www.nesc.ac.uk/technical_papers/UKeS-2004-01.pdf

[7] Coveney, P.V., Harvey, M.J., Pedesseau, L., Mason, D., Sutton, A., McKeown, M. and Pickles, S. Development and deployment of an application hosting environment for grid based computational science (2005). *Proceedings of the UK e-Science All Hands Meeting 2005*, Nottingham, UK, 19-22 September 2005: http://www.allhands.org.uk/2005/proceedings/papers/366.pdf

[8] Beckles, B., Welch, V. and Basney, J. Mechanisms for increasing the usability of grid security (2005). *Int. J. Human-Computer Studies* **63** (1-2) (July 2005), pp. 74-101: http://dx.doi.org/10.1016/j.ijhcs.2005.04.017

[9] Sinnott, R. Development of Usable Grid Services for the Biomedical Community (2006). *Designing for e-Science: Interrogating new scientific practice for usability, in the lab and beyond*, Edinburgh, UK, 26-27 January 2006.

[10] Beckles, B. Re-factoring grid computing for usability (2005). *Proceedings of the UK e-Science All Hands Meeting*

2005, Nottingham, UK, 19-22 September 2005: http://www.allhands.org.uk/2005/proceedings/papers/565.pdf
[11] Pickles, S.M., Blake, R.J., Boghosian, B.M., Brooke, J.M., Chin, J., Clarke, P.E.L., Coveney, P.V., González-Segredo, N., Haines, R., Harting, J., Harvey, M., Jones, M.A.S., McKeown, M., Pinning, R.L., Porter, A.R., Roy, K. and Riding, M. The TeraGyroid Experiment (2004), *Proceedings of GGF10*, Berlin, Germany, 2004: http://www.cs.vu.nl/ggf/apps-rg/meetings/ggf10/TeraGyroid-Case-Study-GGF10-final.pdf
[12] Schneier, B. Secrets and Lies: Digital Security in a Networked World. John Wiley & Sons, 2000.
[13] Beckles, B., Brostoff, S., and Ballard, B. A first attempt: initial steps toward determining scientific users' requirements and appropriate security paradigms for computational grids (2004). *Proceedings of the Workshop on Requirements Capture for Collaboration in e-Science*, Edinburgh, UK, 14-15 January 2004, pp. 17-43: http://www.escience.cam.ac.uk/papers/req_analysis/first_attempt.pdf
[14] Beckles, B. Kerberos: a usable grid authentication protocol (2006). *Designing for e-Science: Interrogating new scientific practice for usability, in the lab and beyond*, Edinburgh, UK, 26-27 January 2006.
[15] Provos, N., Friedl, M. and Honeyman, P. Preventing Privilege Escalation (2003). *12th USENIX Security Symposium*, Washington, DC, USA, 2003: http://www.usenix.org/publications/library/proceedings/sec03/tech/provos_et_al.html
[16] Lock, R., and Sommerville, I. Grid Security and its use of X.509 Certificates. *DIRC internal Conference* submission 2002. Lancaster DIRC, Lancaster University, 2002: http://www.comp.lancs.ac.uk/computing/research/cseg/projects/dirc/papers/gridpaper.pdf
[17] Broadfoot, P. J. and Martin, A. P. A critical survey of grid security requirements and technologies (2003). Programming Research Group Research Report PRG-RR-03-15, Programming Research Group, Oxford University Computing Laboratory, August 2003: http://web.comlab.ox.ac.uk/oucl/publications/tr/rr-03-15.html
[18] Foster, I., Kesselman, C., Tsudik, G. and Tuecke, S. A security architecture for computational grids (1998). *Proceedings of the 5th ACM conference on Computer and communications security*, San Francisco, CA, 1998, pp.83-92: http://portal.acm.org/citation.cfm?id=288111
[19] Basney, J., Humphrey, M. and Welch, V. The MyProxy online credential repository (2005). *Software: Practice and Experience* **35** (9):801-816, 25 July 2005: http://dx.doi.org/10.1002/spe.688
[20] Chadwick, D.W. and Otenko, O. The PERMIS X.509 role based privilege management infrastructure (2002). *Proceedings of the 7th ACM symposium on Access control models and technologies*, Monterey, CA, 2002, pp. 135-140: http://portal.acm.org/citation.cfm?id=507711.507732
[21] Brostoff, S., Sasse, M.A., Chadwick, D., Cunningham, J., Mbanaso, U. and Otenko, S. *R-What?* Development of a role-based access control policy-writing tool for e-Scientists. *Software: Practice and Experience* **35** (9):835-856, 25 July 2005: http://dx.doi.org/10.1002/spe.691
[22] Zurko, M.E. and Simon, R.T., User-centered security (1996). *Proceedings of the 1996 workshop on New security paradigms*, Lake Arrowhead, CA, USA, 1996, pp. 27-33: http://portal.acm.org/citation.cfm?id=304859
[23] Adams, A. and Sasse, M.A. Users are not the enemy: Why users compromise security mechanisms and how to take remedial measures (1999). *Communications of the ACM*, Volume 42, Issue 12 (December 1999), pp. 40-46: http://portal.acm.org/citation.cfm?id=322806
[24] Gould, J.D. and Lewis, C. Designing for Usability: Key Principles and What Designers Think (1985). *Communications of the ACM*, Volume 28, Issue 3 (March 1985), pp. 300-311: http://portal.acm.org/citation.cfm?id=3170
[25] Beyer, H. and Holtzblatt, K. Contextual Design: Defining Customer-centered Systems. Morgan Kaufmann Publishers, 1998.
[26] Cooper, A. The Inmates Are Running the Asylum: Why High-tech Products Drive Us Crazy and How to Restore the Sanity. Sams, 1999.
[27] Snelling, D.F., van den Berghe, S. and Li, V. Q. Explicit Trust Delegation: Security for Dynamic Grids. *FUJITSU Sci. Tech. J.* **40** (2) (December 2004), pp. 282-294: http://www.unigrids.org/papers/explicittrust.pdf
[28] Beckles, B. Removing digital certificates from the end-user's experience of grid environments (2004). *Proceedings of the UK e-Science All Hands Meeting 2004*, Nottingham, UK, 31 August - 3 September 2004: http://www.allhands.org.uk/2004/proceedings/papers/250.pdf
[29] Pearlman, L., Welch, V., Foster, I., Kesselman, C., Tuecke, S. A Community Authorization Service for Group Collaboration (2002). *Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, Monterey, CA, 2002, pp. 50-59: http://www.globus.org/alliance/publications/papers/CAS_2002_Revised.pdf
[30] Alfieri, R., Cecchini, R., Ciaschini, V., dell'Agnello, L., Frohner, A., Gianoli, A., Lorentey, K.L., Spataro, F.. VOMS, an Authorization System for Virtual Organizations (2003). 1st European Across Grids Conference, Santiago de Compostela, Spain, 13-14 February 2003: http://grid-auth.infn.it/docs/VOMS-Santiago.pdf
[31] Shibboleth Project: http://shibboleth.internet2.edu/
[32] Van, T. "Grid Stack: Security debrief". *Grid Stack*, 17 May 2005: http://www-128.ibm.com/developerworks/grid/library/gr-gridstack1/index.html
[33] Yee, K-P. User interaction design for secure systems (2002). *Proceedings of the 4th International Conference on Information and Communications Security*, Singapore, 9-12 December 2002. Expanded version available at: http://zesty.ca/sid
[34] Balfanz, D., Durfee, G., Grinter, R.E, Smetters, D.K. In search of usable security: Five lessons from the field (2004). *IEEE Security and Privacy* **2** (5) (September-October 2004), pp. 19-24: http://doi.ieeecomputersociety.org/10.1109/MSP.2004.71
[35] Fléchais, I., Sasse, M.A. and Hailes, S.M.V. Bringing Security Home: A process for developing secure and usable systems (2003). *Proceedings of the 2003 workshop on New security paradigms*, Switzerland, 2003, pp. 49-57: http://portal.acm.org/citation.cfm?id=986664
[36] Cooper, A., and Reimann, R. About Face 2.0: The Essentials of Interaction Design. John Wiley and Sons, New York, 2003.
[37] Sommerville, I. An Integrated Approach to Dependability Requirements Engineering (2003). *Proceedings of the 11th Safety-Critical Systems Symposium*, Bristol, UK, 2003.
[38] Ryan, P.Y.A., Schneider, S.A., Goldsmith, M.H., Lowe, G., and Roscoe, A.W. Modelling and Analysis of Security Protocols. Addison-Wesley Professional, 2000.
[39] Abdallah, A.E., Ryan, P.Y.A. and Schneider, S. (eds.). Formal Aspects of Security. Proceedings of the First International Conference, FASec 2002, London, December 2002. LNCS 2629, Springer-Verlag, 2003.
[40] RealityGrid: http://www.realitygrid.org/
[41] Beckles, B. User requirements for UK e-Science grid environments (2004). *Proceedings of the UK e-Science All Hands Meeting 2004*, Nottingham, UK, 31 August - 3 September 2004: http://www.allhands.org.uk/2004/proceedings/papers/251.pdf
[42] Snyder, C. Paper prototyping: The fast and easy way to design and refine user interfaces. Morgan Kaufmann Publishers, London, 2003.

A virtual research organization enabled by *e*Minerals minigrid: an integrated study of the transport and immobilization of arsenic species in the environment

Z. Du[1,2], V.N. Alexandrov[11], M. Alfredsson[3], E. Artacho[6], K.F. Austen[6] , N.D. Bennett[8], M. Blanchard[4], J.P. Brodholt[3], R. P. Bruin[6], C.R.A. Catlow[4], C. Chapman[10], D.J. Cooke[5], T.G. Cooper[9], M.T. Dove[6,7], W. Emmerich[10], S.M. Hasan[11], S. Kerisit[5], N.H. de Leeuw[1,2] , G.J. Lewis[11], A. Marmier[5], S.C. Parker[5], G.D. Price[3], W. Smith[8], I. T. Todorov [8], R.P.Tyer[8] , K Kleese van Dam[8], A.M. Walker[6], T.O.H. White[6], K. Wright[4]

1. School of Crystallography, Birkbeck College, Malet Street, London, WC1E 7HX
2. Department of Chemistry, University College London, London WC1H 0AJ
3. Department of Earth Sciences, University College London, London WC1E 6BT
4. Royal Institution of Great Britain, Albemarle Street, London W1S 4BS
5. Department of Chemistry, University of Bath, BA2 7AY
6. Department of Earth Sciences, University of Cambridge, Downing Street, Cambridge CB2 3EQ
7. National Institute for Environmental *e*Science, Centre for Mathematical Sciences, University of Cambridge, Wilberforce Road, Cambridge CB3 0WA
8. CCLRC, Daresbury Laboratory, Daresbury, Warrington, Cheshire WA4 4AD
9. The Pfizer Institute for Pharmaceutical Materials Science, Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW
10. Department of Computer Science, University College London, London WC1E 6BT
11. Department of Computer Science, University of Reading, PO Box 225, Reading RG6 6AY

Abstract

We have carried out a comprehensive computational study of the structures and properties of a series of iron-bearing minerals under various conditions using grid technologies developed within the *e*Minerals project. The work has enabled by a close collaboration between computational scientists from different institutions across the UK, as well as the involvement of computer and grid scientists in a true team effort. We show here that our new approach for scientific research is only feasible with the use of the *e*Minerals minigrid. Prior to the *e*Minerals project, such an approach would have been almost impossible within the timescale available. The new approach allows us to achieve our goals in a much quicker, more comprehensive and detailed way. Preliminary scientific results of an investigation of the transport and immobilization of arsenic species in the environment are presented in the paper.

1. Introduction

The remediation of contaminated waters and land poses a grand challenge on a global scale. The contamination of drinking water by arsenic has been associated with various cancerous and non-cancerous health effects and is thus one of the most pressing environmental concerns [1]. It is clearly important to understand the mechanisms of any processes that reduce the amount of active arsenic species in groundwater, which can be achieved by selective adsorption. It is generally considered that iron-bearing minerals are promising absorbents for a wide range of solvated impurities. There is a serious need for a detailed understanding of the capabilities of different iron-bearing minerals to immobilize contaminants under varying conditions, in order to guide strategies to reduce arsenic contamination. We have chosen to use a computational approach to study the capabilities of iron-bearing minerals to immobilize active arsenic species in water.

This large, integrated project requires input from personnel with diverse skills and rich experience, as well as techniques and infrastructure to support a large scale computing effort. The complex calculations required include workflows, high-throughput and the ability to match different computational need against available resources. Furthermore, data management tools are required to support the vast amount of data produced in such a study. This is clearly a real challenge for the *e*-science

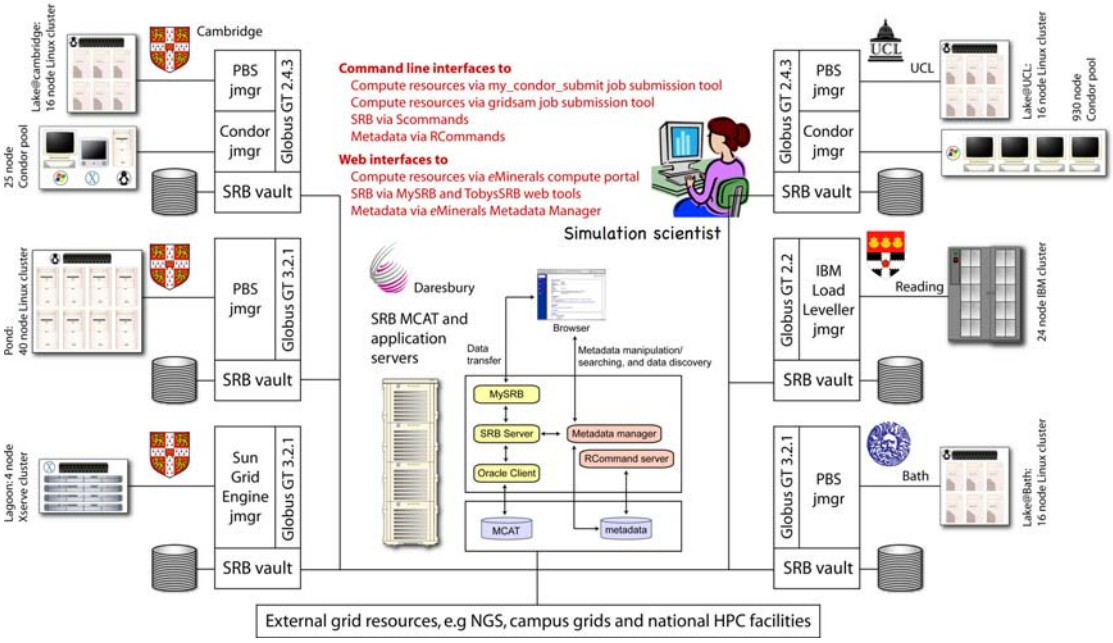


Figure 1 Components of *e*Minerals minigrid. It contains heterogeneous computer resources: 3 Linux-based clusters situated at Bath, Cambridge and UCL; 1 Linux-cluster with 8GB per node at Cambridge; 1 24-node IBM parallel machine at Reading; 3 Condor-pools located at UCL, Cambridge and Bath. Besides, external facilities *e.g.* NGS, CSAR and HPCx are also accessible. The Storage Resource Broker (SRB) enables data stored at various sites to be seen by the users as if within one single file.

community. The aim of the *e*Minerals project [2] is to incorporate grid technologies with computer simulations to tackle complex environmental problems. During the first phase of this project, *e*Minerals minigrid has been developed and tested. In addition, a functional virtual organisation has been established, based on the sharing of resources and expertise, with supporting tools.

We can now fully explore the established infrastructure to perform simulations of environmental processes, which means having to deal with very complicated simulation systems close to 'real systems' and under varying conditions. Here we report on a case study using the grid technologies developed in the *e*Minerals project to conduct a large integrated study of the transport and immobilization of arsenic in a series of iron-bearing minerals, where the computational scientists work as a team to tackle one common problem. The whole team operates as a task-oriented virtual organization (VO) and each member of the VO will use his own expertise to deal with one or several aspects of the project. A close collaboration between all team members is inevitable in such a comprehensive study, which enables the scientific studies to be carried out at different levels, using different methodologies and investigating many different iron-bearing minerals. This new approach is aimed at expanding the horizons of simulation studies and meeting the needs for a quick, comprehensive and detailed investigation into the mechanisms of arsenic adsorption by various adsorbents. The investigation at such a scale would be impossible without the collaborative grid technologies.

2. Grid technologies in computational simulations

All the simulations conducted in this study were facilitated by the *e*Minerals minigrid [3]. The eminerals mini-grid is composed of a collection of shared and dedicated resources. These are spread across 6 sites, and are depicted in figure 1. We also regularly complement these resources with external resources such as those made available by the National Grid Service. More details of building and examining tools as well as our view and experience on using these tools have been discussed in ref [3,4]. Here we just give some brief description.

2.1 Communication tools

As mentioned in the introduction, this project is a large collaboration in terms of the number of people involved and the levels of calculations

needed to be carried out. Good organization and communication is therefore the key to the institutions across the UK, it would be extremely difficult to keep the whole team in communication on a daily or weekly basis if we followed traditional methods.

Within the *e*Minerals project, we have developed a new approach, taking advantage of different communication tools like: Personal Access Grid (AG) and a *Wiki,* Instant message (IM) in addition to more conventional tools (e.g. e-mails, telephone). Each tool is suitable to a particular communication required by the collaboration.

- **AG meeting:** AG does offer a valuable alternative to the expense of travelling for project meetings. The AG meetings are now acting as a valuable management tool and in the context of the project enabling us to identify the most suitable person to complete the selected simulations within a specified time. For instance, for the arsenic pollutant problem, through the AG meeting, we have decided that team members from Bath and Birkbeck focus on adsorption of arsenic species on iron oxides and hydroxides using interatomic potential based simulations, while those from the Royal Institution and Cambridge work on arsenic species adsorption on pyrite and dolomite using quantum mechanical calculations. The AG meetings also play an important role in members contributing their ideas to collaborative papers like this one.
- **The *Wiki*:** *Wiki* is a website which can be freely edited by the members of the VO. We have use *Wiki* to exchange ideas, deposit news information, edit collaborative papers. For example, we have used Blogbooks for the resource management to notify other members before submitting large suites of calculations. The downside of this tool is that *wiki* does not support instant communication, which limits the quality of its use as a collaborative tool.
- **Instant message (IM):** IM gives much better instantaneous chat facilities than email and *wiki*. It is particular useful for members of the project developing new tools because it is easy to ask questions and reach agreement.

2.2 Grid environment support for workflow

All calculations conducted in this study have been organized by a meta-scheduling job submission tool called my_condor_submit success of the project. Because the members of the project are based at different (MCS) [5]. The MCS manages the simulation workflow using Condor's DAGman functionality and integrates with Storage Resource Brokers (SRB) for data management. The Condor interface has proved quite popular, and facilitates the submission and management of large numbers of jobs to globus resources. In addition, it enables the use of additional Condor tools and in particular the DAG-Man workflow manager. SRB has provided seamless access to distributed data resources. As shown in figure1, there are 5 SRB data storage vaults spread across 4 sites within the project amounting to a total capacity of roughly 3 Terabytes. A metadata annotation service (RCommands service)[6] hosted by the CCLRC, and database back end, was used to generate and search metadata for files stored in the SRB vaults.

The use of MCS together with SRB allows us to handle each simulation following a simple and automated three-step procedure. The input files are downloaded from the SRB and MCS decides where to run the job within the *e*Minerals minigrid, depending on the available compute facilities. Finally the output files are uploaded back to the SRB for subsequent analysis. This tool certainly facilitates the work of the scientists and avoids unnecessary delays in the calculations.

Our practice has shown that the SRB is of prime importance for data management in such collaboration. It not only provides a facility to accommodate a large number of data sets, but also allows us to share information more conveniently, e.g. large files that are nigh impossible to transfer using traditional communication tools (e.g. email) due to size limitations. SRB also permits us to access the data files wherever and whenever we wish.

Here we just illustrate a couple of examples to show the benefits of using grid techniques to support workflow in a scientific study. In the case of the quantum mechanical study of the structures of goethite, pyrite and wüstite, we have to take into account that these minerals can have different magnetic structures, which requires a range of separate calculations. For example, ferric iron, which has five d-electrons, exists with magnetic numbers equal to 5 (HS) and 1 (LS), while ferrous iron, which has 6 d-electrons, can have a magnetic number 4 (HS) or 0 (LS). This means that the minerals with ferrous iron can have anti-ferromagnetic (AFM); ferromagnetic (FM); and non-magnetic (NM) structures, while minerals containing ferric iron show AFM or FM structures. For

Table 1 Electronic and magnetic structures as predicted by the hybrid-functionals reported in the text as well as the PW91 (GGA) Hamiltonian and experiments. HS=high spin, LS=low spin, M= metal and I=insulator.

Mineral	hybrid	Exp	GGA	hybrid	GGA	Exp
FeO	HS	HS[9]	HS	I	M	I[14]
FeS_2	LS	LS [10]	LS	I	M	I[15]
$FeSiO_3$	HS	-	HS	I	I	-
Fe_2SiO_4	HS	HS[11]	LS	I	M	I[16]
Fe_2O_3	HS	HS[12]	LS	I	M	I[17]
α-FeOOH	HS	HS[13]	HS	I	I	I[18]

each mineral we consider 5 to 10 different hybrid-functionals for several magnetic structures, which means that we run 20 to 30 compute intensive calculations per mineral. As all calculations are independent of each other, they are carried out on the UCL Condor-pool (> 1000 processors) using the MCS job submission tool. As such, the calculations can start almost immediately and are completed within a couple of months, whereas, prior to this eScience technology, this type of study might have taken a year or longer to conclude.

As another example, the study of arsenic incorporation in FeS_2 pyrite has required accurate geometry optimizations of supercells at a quantum mechanical level. Due to the nature of these calculations, the national computing facility (HPCx) was used. In addition, many smaller and independent calculations were carried out for convergence tests and especially for obtaining the total energy of all the different reaction components. This task was suited perfectly to the coupled use of the *e*Minerals minigrid and SRB via the MCS tool.

3. Science outcomes

One of the major environmental concerns in recent years has been the widespread contamination by arsenic of aquifers used for drinking water, for example in Bangladesh. It is clearly of immense importance to reduce the presence of active arsenic species in the water and attention is focused on the sorption of these and other contaminants onto minerals, which are present in the environment of the aquifers.

The common goal of this project is to investigate the transport and immobilisation mechanisms of arsenic species in various iron-bearing minerals, which are thought to be promising adsorbents [7]. We report here our preliminary results, namely, the theoretical description of the bulk minerals, surface stabilities and hydration processes.

3.1 Quantum mechanical studies of the structures of Goethite, Pyrite and Wüstite

The modelling of iron-bearing minerals is a particular challenge in mineral physics as the electronic and magnetic structures of many of these minerals are badly or even wrongly represented by traditional density functional theory (DFT). Most of the minerals we have studied are predicted to be metals; although experimentally they are reported as insulators (see Table 1 for details). In addition, DFT gives the incorrect spin state for Fe_2SiO_4 and Fe_2O_3 when the GGA functional is used, thus give rise to the wrong magnetic structures of the minerals.

Our aim is to compare GGA and hybrid-functional calculations of a range of iron-bearing minerals to determine if the minerals are well described within the DFT formalism or if they need to be described by more advanced quantum mechanical techniques, e.g. hybrid-functionals. Here we have chosen to discuss three minerals in more detail: a) Goethite (α-FeOOH), b) Pyrite (FeS_2) and c) Wüstite (FeO).

a) Goethite

In goethite the Fe-species are in oxidation state (III) (ferric-iron), and as shown in Table 1 we predict both the magnetic and electronic structures correctly with both GGA and hybrid-functionals for this mineral. However, the lattice parameters optimised with GGA are in better agreement with experiment than the values obtained using the hybrid-functionals. Comparing the internal geometries we find that both the hybrid-functionals as well as the GGA Hamiltonian give values in agreement with experiment. Both computational techniques on the other hand underestimate the internal

O-H bond distance and the hydrogen bond distance. Our calculations suggest that α-FeOOH is adequately described in DFT and there is no need for alternative techniques to study this material.

b) Pyrite

Pyrite contains ferrous iron, and from Table 1 we see that DFT fails to predict this mineral as insulator. However, the magnetic structure for pyrite (LS) agrees with experiment. Pyrite has also been studied using plane-waves to describe the electron density (e.g. ref. 8), while we are using localised Gaussian-basis functions. In the previous study pyrite is correctly reported as an insulator. We also find that both the hybrid-functionals and GGA overestimate the lattice parameters compared to the experimental value. We believe this is again due to the description of the electron density by the Gaussian-basis functions, which can be improved. Hence, our conclusion from this study is that pyrite can be described by plane-wave DFT methods and does not need to be described by hybrid-functionals.

c)Wüstite

FeO is a classical example where DFT is known to fail to predict the electronic structure, as shown in Table 1. If we concentrate on the structural properties of wüstite on the other hand we find that GGA underestimates, while the hybrid-functional described by 40% Hartree-Fock exchange overestimates the lattice parameters. This is also reflected in the internal geometries. Hence, our conclusions are that wüstite is better described by hybrid-functionals, but some properties, like the geometrical properties are well described with GGA. However, the magnetic and electronic structures require hybrid-functionals to be correctly described.

Our study showed that minerals containing ferric iron are often better described within DFT than minerals containing ferrous iron. However, ferrous iron in LS configurations, such as pyrite, is also well described within DFT, while minerals containing ferrous iron, which show magnetism (i.e. HS configuration) are better described by hybrid-functionals. This result is a reflection of the known shortcomings of DFT and its representation of electron correlation; high-spin ionic states are always poorly represented within the DTF approximation, so we expect the structural features of HS ferric iron to be captured considerably worse, and requiring the additional accuracy provided by hybrid functional approaches, while for LS ferrous iron, DFT alone gives a reasonable result.

3.2 Arsenic incorporation in pyrite FeS_2

Pyrite (FeS_2) the most abundant of all metal sulphides, plays an important role in the transport of arsenic. Under reducing conditions, pyrite can delay the migration of arsenic by adsorption on its surfaces, as well as by incorporation into the bulk lattice. Pyrite can host up to about 10 wt % of arsenic [19]. Under oxidizing conditions, pyrite dissolves, leading to the generation of acid drainage and releasing arsenic into the environment. Although it is key information, the location and speciation of arsenic in pyrite remains a matter of debate. The pyrite has a NaCl-like cubic structure with an alternation of Fe atoms and S_2 groups. X-ray adsorption spectroscopic studies showed that arsenic substitutes for sulphur, forming AsS di-anion groups rather than As_2 groups [20]. On the other hand a recent experimental study has proposed that arsenic can also act like a cation, substituting for iron. The different arsenic configurations have been investigated using first-principles calculations.

Calculations have been executed within the DFT framework. In order to work with arsenic concentrations comparable with natural observations, calculations were performed on 2x2x2 pyrite supercells containing up to two arsenic atoms (< 4 wt. % As). The arsenic has been successively placed in iron and sulphur sites. In the later case, we have investigated the three following substitution mechanisms: formation of AsS groups, formation of As_2 groups and substitution of one As atom for one S_2 group. These different configurations have been compared by considering simple incorporation reactions under both oxidizing and reducing conditions. Solution energies suggest that, in conditions where pyrite is stable, arsenic will preferentially be incorporated in a sulphur site forming AsS groups (Fig 2). The incorporation of arsenic as a cation is energetically unfavourable in pure FeS_2 pyrite. Previous studies have shown that above an arsenic concentration of about 6 wt%, the solid solution becomes metastable and segregated domains of the arsenopyrite are formed in the pyrite lattice [21]. Even for the low concentrations modelled here, the substitution energies indicate that the arsenic tends to cluster. Our results also show that, in oxidising conditions, the presence of arsenic

.

Table 2 Surface energies for iron (hydr)oxide minerals including both dry and hydroxylated surfaces

Fe_2O_3									
Surface	001	012a	012b	012c	100	101	110	111a	111b
Dry surface energy (Jm^{-2})	1.78	1.87	2.75	2.35	1.99	2.34	2.02	2.21	2.07
Hydrated surface energy (Jm^{-2})	0.90	0.38	0.28	0.38	0.27	0.04	0.20	0.33	0.28

FeOOH							
Surface	010	100	110	001	011	101	111
Dry surface energy(Jm^{-2})	1.92	1.68	1.26	0.67	1.18	1.72	1.33
Hydrated surface energy (Jm^{-2})	0.51	1.17	0.68	0.52	0.34	1.32	1.12

$Fe(OH)_2$						
Surface	001	010	011	101	110	111
Dry surface energy (Jm^{-2})	0.04	0.38	0.35	0.35	0.64	0.60

may accelerate the dissolution of pyrite with the en environmental consequences that implies (acid rock drainage and release in soluaiton of toxic metals).

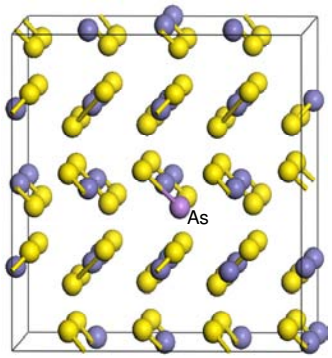


Figure 2 Relaxed structure of the 2x2x2 supercell of pyrite containing one AsS group.

3.3 Iron (hydr)oxide mineral surfaces

Iron (hydr)oxides play an important role in many disciplines including pure, environmental and industrial chemistry, soil science, and biology. Iron (hydr)oxide minerals are thought to be promising adsorbents to immobilize active arsenic and other toxic species in groundwater. There are eleven known crystalline iron (hydr)oxide compounds and the focus of our work here is initially on three representative systems, namely the pure hydroxide $Fe(OH)_2$, the mixed oxyhydroxide goethite FeOOH, and the iron oxide hematite Fe_2O_3.

In order to investigate the adsorption mechanisms of arsenic species on the iron (hydr)oxide minerals, we have first carried out a large number of interatomic potential-based simulations to examine the surface structures and stabilities of these minerals. We have presented the calculated surface energies for both dry and hydrated surfaces are listed in Table 2

The calculated surface energies of the three minerals show that in the case of dry Fe_2O_3, the iron terminated {001} surface is the most stable surface whereas the {012b} surface is the least stable one. Surface structure analysis indicated that the atoms on the {001} surface have relatively high coordination compared to other surfaces, which makes this surface less reactive. In general, the surface energies of the hydroxylated surfaces are lower than those of the corresponding dry surfaces. Upon hydration, the iron-terminated {001} surface becomes the least stable surface, although the reconstructed dipolar oxygen-terminated surface now becomes highly stable, due to the adsorbed water molecules filling oxygen vacancies at the surface.

Like hematite, the hydrated surfaces of goethite are more stable than the dry ones. In $Fe(OH)_2$, the surface energies for all surfaces are lower compared to those of Fe_2O_3 and FeOOH, which is mainly due to the fact that $Fe(OH)_2$ is an open layered structure.

Overall our calculated energies as well as structure analysis indicate that all the surfaces where the surface atoms are capped by OH groups have relatively low surface energies, which implies that the presence of hydroxyls groups helps to stabilise the surface.

3.4 Aqueous solutions in the vicinity of iron hydroxide surfaces

Although environmentally relevant immobilisation processes concern the adsorption from solution, the exact structure of aqueous solutions in contact with surfaces is not yet completely elucidated. The distribution and local concentration of the various species is difficult to observe experimentally. Coarse grained Surface Complexation Models do not include explicitly surface effects, but conversely, ultra precise *ab initio* calculations are unable to cope with the amount of water required. The intermediate solution, the use of classical atomistic modelling techniques, was also traditionally hampered by the fact that realistic ionic concentrations require the treatment of many water molecules (at a factor of 50 water molecules per ion at the already high concentration of 1 mol l^{-1}).

However, developments in computer power have now made it possible for simulations of both surface and solution at atomic resolution and in large enough quantity to produce statistically meaningful results. We have carried out many Molecular Dynamics simulations (DL_POLY code) of (Na^+/Cl^-)/ goethite interfaces, in an aqueous solution at different ionic strengths and surface charges. Our main observation is that the distribution of ions near the surface is not accurately described by the classical models of the electrical double layer.

It is considered that any surface has a structuring effect on the liquid it is in contact with, as can be observed in the density oscillations in figure 3. This density rippling in turn controls the salt concentrations. There is a direct correlation with the water density up to 10Å from the surface, but relatively unexpected,

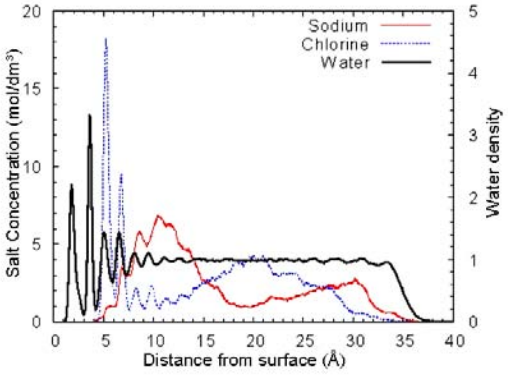


Figure 3 Salt concentrations and water density dependence on the distance from the surface of goethite.

broad, long-range oscillations continue further away. The explanation lies in the electrostatic potential, as pictured in figure 4. The structuring effect of the surface on the electrostatic potential of pure water has a longer range than what could be inferred from the simpler density curves. The addition of ions in solution only serves to reinforce this effect.

Real surfaces are likely to be charged. But the corresponding simulations show that the effect of the charged surface on the electrostatic potential in the solvent does not significantly differ from the neutral surfaces.

Additional calculations of surfaces of different minerals (calcite $CaCO_3$ and hematite Fe_2O_3) confirm these findings and suggest that the long range oscillatory behaviour of salt concentration is a consequence of the structuring presence of a surface on the electrostatic potential of the water.

We conclude that although the traditional double (stern)-layer models are correct in assuming that the ion distribution is controlled by the electrostatic potential, they fail to reproduce the distribution at medium/high salt concentration because the electrostatic contribution of the structured (layered) solvent is not taken into account.

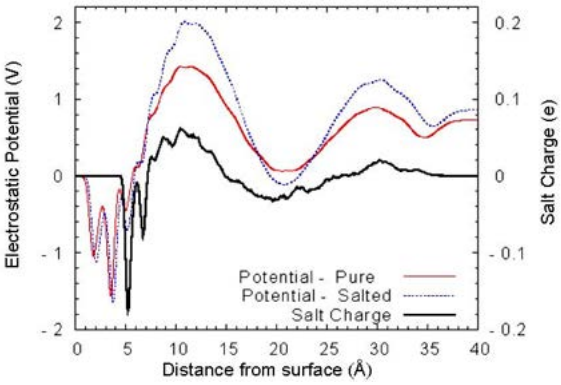


Figure 4 Electrostatic potential (in pure and salted solution) and charge dependence on the distance from the surface of goethite.

4. Conclusions and future work

Modelling of complex processes that occur in the environment is extremely challenging since simulations of large systems are required and complicated by the effects of many physical and chemical factors (e.g. T, P, pH). Although this large collaborative work is still in its early stages, it has already shown its promise to solve complex scientific problems with the support of grid technologies.

In the near future, we will extend the study to investigate the immobilization of a number of arsenic species by various iron-bearing minerals. A large number of calculations will be carried out to investigate the adsorption mechanisms of arsenic species of different oxidation state (e.g. As(III) and As(V)) on different iron-bearing mineral surfaces as well as studying the influence of varying conditions on the adsorption process. The outcome of the project will be a detailed atomic-level understanding of the chemical and physical processes involved in the immobilization of arsenic species by a variety of important iron minerals, which will benefit a wide range of communities, such as water treatment and environment agencies; academic and industrial surface scientists; mineralogists and geochemists.

Acknowledgements
This work was funded by NERC via grants NER/T/S/2001/00855**,** NE/C515698/1 and NE/C515704/1.

References

[1] D. J. Vaughan, "Arsenic", *Elements*, 2 71(2006).
[2] M. T. Dove and N. H. Leeuw, "Grid computing and molecular simulations: the vision of the *e*Minerals project", *Mol. Sim.*, **31**, 297 (2005).
[3] M. Calleja et al., "Collaborative grid infrastructure for molecular simulations: The *e*Minerals minigrid as a prototype integrated computer and data grid", *Mol. Sim.*, **31**, 303 (2005).
[4] M. T. Dove et al., "The eminerals collaboratory: tools and experience", *Mol. Sim.*, **31**, 329 (2005).
[5] R.P. Bruin, et al., "Job submission to grid computing environments", *All Hands Meeting* (2006).
[6] R.P. Tyer et. al., "Automatic metadata capture and grid computing", *All Hands Meeting* (2006).
[7] K.G. Stollenwerk, "Geochemical processes controlling transport of arsenic in groundwater: A review of adsorption". In: A.H. Welch, K. G. Stollenwerk (eds) Arsenic in groundwater: Geochemistry and Occurrence, Kluwer Academic publishers, Boston, pp67-100 (2003).
[8] M. Blanchard, M. Alfredsson, J.P. Brodholt, G.D. Price, K. Wright and C.R.A. Catlow, "Electronic structure study of the high-pressure vibrational spectrum of FeS2 pyrite". *J. Phys. Chem. B*, **109**, 22067 (2005).
[9] N.C. Tombs and H.P. Rooksby, "Structure of monoxides of some transition elements at low temperatures", *Nature*, **165**, 442 (1950).
[10] R.W.G. Wyckoff, *Crystal structures*, **1**, 346 (1963).
[11]K. Fujino, S. Sasaki, Y. Takeuchi and R.Sadanaga, "X-ray determination of electron distributions in forsterite, fayalite and tephroite", *Acta Cryst.*, **B37**, 513 (1981).
[12] R.L. Blacke, R.E. Hessevick, T. Zoltai, L.W. Finger, "Refinement of hematite structure", *Amer. Mineral.*, **51**, 123 (1966).
[13] A.F. Gaultieri and P. Venturelli, "In situ study of the goethite-hematite phase transformation by real time synchrotron powder diffraction", *Amer. Mineral.*, **84**, 895 (1999).
[14] P.S. Bagus, C.R. Brundle, T.J. Chuang and K. Wandelt, "Width of D-level final-state structure observed in photoemission spectra of fexo", *Phys. Rev. Lett*., **83**, 1229 (1977).
[15] I. Opahle, K. Koepernik and H. Escrig, "Full-potential band-structure calculation of iron pyrite", *Phys. Rev. B*, **60**, 14035 (1999).
[16] Q. Williams, E. Knittle, R. Reichlin, S. Martin and R. Jeanloz, "Structural and electronic-properties of Fe_2 SiO_4 –fayalite at ultrahigh pressures-amorphization and gap closure", *J. Geophys. Res. [Solid Earth]* **95**, 21549 (1990).
[17] A. Fujimori, M. Saeki, N. Kimizuka, M. Taniguchi and S. Suga, "Photoemission satellites and electronic-structure of Fe_2O_3", *Phys. Rev. B*, **34**, 7318 (1986).
[18] M. Ziese and M.J. Thornton *Materials for spin electronics*, (eds), Springer, (2001).
[19] P.K. Abraitis, R.A.D. Pattrick, D.J. Vaughan "Variations in the compositional, textural and electrical properties of natural pyrite: a review". *Int. J. Miner. Process.,* **74,** 41 (2004).
[20] K.S. Savage, T.N. Tingle, P.A. O'Day, G.A. Waychunas and D.K. Bird, "Arsenic speciation in pyrite and secondary weathering phases", *Appl. Geochem.,* **15**, 1219 (2000).
[21] M. Reich and U. Becker, "First-principles calculations of the thermodynamic mixing properties of arsenic incorporation into pyrite and marcasite". *Chem. Geol.,* **225,** 278 (2006).

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

The GOLD Project: Architecture, Development and Deployment

Hugo Hiden[1], Adrian Conlin[1], Panayiotis Perrioellis[1], Nick Cook[1], Rob Smith[1], Allen Wright[2]

[1]Department of Computing Science

[2] Department of Chemical Engineering and Advanced Materials

University of Newcastle upon Tyne

Abstract

This paper presents a description of the architecture, development and deployment of the Middleware developed as part of the GOLD project. This Middleware has been derived from the requirement to accelerate the chemical process development lifecycle through the enablement of highly dynamic Virtual Organisations. The generic design of the Middleware will allow its application to a wide variety of additional domains.

1. Introduction

GOLD is an EPSRC funded e-Science pilot project which aims to accelerate the Chemical Process Development (CPD) lifecycle through the enablement of Virtual Organisations (VOs) (Demchenko, 2004) and active Information Management. This complex application domain has two dominant characteristics, which have not been explored by previous Grid research.

Extremely dynamic virtual organisations: The chemical R&D lifecycle is highly dynamic and unanticipated direction changes may occur at any point. Agility and flexibility are essential to respond to these changes and ensure time to market is minimised. The entire workflow for developing a given product will not be known at the outset. A need for different and or additional outsourcing of specialist services may become apparent as project knowledge increases. For example, additional resource required to prevent undesirable environmental impact of by-products identified during the project. This type of variability within and across projects demands a highly flexible outsourcing model for the VOs. Binding to specific organisations or services in a given project must occur at the latest possible moment.

Full lifecycle focus: The project seeks to integrate the full lifecycle of CPD, from basic research through design and process engineering to manufacture. In many product development cycles these phases are operated separately in distinct divisions or by separate companies. There are potentially a wide range of classes of interaction between VO participants during a chemical development project. These range from exchange of basic design data to specifying and ordering physical plant equipment. Some of these interactions, particularly those involving orders for equipment and manufacturing time, need to be non-repudiable (Cook, et al, 2002) in case of future disputes.

CPD requires the exchange of wide range of information class between VO participants. This information ranges from physical property

data, laboratory notes, experimental data, safety studies during the initial development stages through to industrial plant design information if and when the projects reach the commercial exploitation phase.

Much of the information exchanged between participants is confidential, and must be secured from unauthorised access. The security requirements are demanding because the access control must change over the course of the development process, allowing different levels of access to certain individuals as projects progress.

2. Chemical Development Scenario

The functional requirements of the GOLD middleware have been identified from a number of sources including industrial consultation and the extensive CPD experiences of some of the GOLD team members. In addition an actual CPD project has been undertaken in collaboration with a number of companies as part of the development of the GOLD demonstrator.

The aim of this ongoing CPD project is to convert an existing high tonnage manufacturing process from batch to continuous operation. In order to accomplish this goal a range of specialist skills are required some of which are outlined in Table 1.

Participant	***Skill***
Reaction Engineering Consulant	Analyses chemical reactions, and suitable operating conditions.
Pilot Plant Designers	Can design, build and operate small scale chemical plants.
Equipment Vendor	Supplies off the shelf process equipment according to supplied specifications.
Equipment Manufacturer	Supplies custom build equipment not available from

Table 1: Participant Roles

A VO approach has been adopted because these skills were not available in either the initiating company or a single contractor. In addition the VO approach has the potential to offer substantial cost savings.

The project consists of four basic phases: Preliminary reaction engineering investigations, pilot plant design, pilot plant operations and commercial scale plant development. Although superficially this is a straightforward project, each of the phases described above can be subject to a number of disturbances leading to significant changes in the subsequent tasks to be performed. For example, the new operating conditions may unexpectedly affect the downstream recovery of the catalyst and a new separation method must be found. This could involve a modification to the VO structure through the introduction of a new specialist skill and or removal of an existing participant. This example is one of many unanticipated factors that could require radical changes to the project plan after the project has been started. The software, therefore, must be able to co-ordinate activities between VO participants in the face of such disturbances.

During the course of the CPD project, significant quantities of information are exchanged between participants. This information usually takes the form of “dossiers”, each of which can contain a set of individual documents. Dossiers can cover various aspects of the development lifecycle, but the scenario considered during the development of GOLD is focused on four main areas: Commercial, Technical, Manufacturing and Safety, Health and Environment (SHE). Access by users to the information contained within these dossiers by VO participants is controlled depending on the roles held by these individuals.

This project has provided a detailed case study outlining tasks performed by and information exchanged between VO participants. Whilst not exhaustive it does provide a reasonably thorough test for the middleware developed by the project.

3. Software Architectural Elements

An examination of the scenario presented above, performed during the software design process (Conlin, et al, 2005) has identified a number of high level architectural elements that the middleware developed by the GOLD project must be able to provide in order to support the scenario described above. A further decomposition of these elements was then performed to identify a number of atomic services that were then implemented in order to demonstrate the application. These elements are broadly classified as:

Storage: Support for storing and retrieving any of the various information types generated and exchanged during the lifecycle of the chemical development process. Also included within this aspect of the architecture is a comprehensive Information Model which describes the various data types and VO structural information stored during the operation of the VO.

Security: Services and facilities required in order to control access to information held withn the VO. These are important because the chemical and pharmaceutical industries attach considerable importance to the security of their information.

Co-ordination: Functionality to enable the activities of individual VO participants to be co-ordinated and performed in accordance with the overall plan for the CPD process currently in progress.

Regulation: Monitors interactions between participants to ensure agreed behaviour and to enable actions performed to be audited at a later date.

Detailed descriptions of theses architectural elements are available in Conlin, et al, 2005, which also describes the various services required in order to support this architecture.

4. Services to Support VOs

The GOLD Middleware has been implemented in the form of Web Services (Skonnard, 2002). The provision of the core software components as services allows VOs to be constructed using a subset of the full provided functionality if required. For example, certain VOs may not require extensive auditing or regulatory functions.

5. Storage Services

The storage services provided to the VO enable all of the information generated during the operation of the VO to be archived and retrieved. In addition to the CPD specific documents, the information includes details of the project plan, the membership of the VO, security attributes etc. In order to support this storage, a unified Information Model has been developed, which is summarised below. The Information Model, which has been based in part upon the MyGrid Information Model (Sharman, et al, 2004) is provided within the software implementation as a Java class hierarchy. Within this hierarchy, there are three base classes:

Class	***Description***
`GoldObject`	The base class for the majority of documents stored within the information repository.
`LogMessage`	Base class for all auditing and non-repudiation log messages.
`GoldDocument`	Represents the actual data from a document stored within the VO information repository. Document indexing and description data is held in the separate `DocumentRecord` class.

Table 2: Information Model Base Classes

Within the Information Model, the `GoldObject` class hierarchy contains details of most of the VO structural data. People, Roles, Participant Companies, VO Projects etc are all subclasses of `GoldObject`. A subset of this hierarchy is shown below in Figure 1.

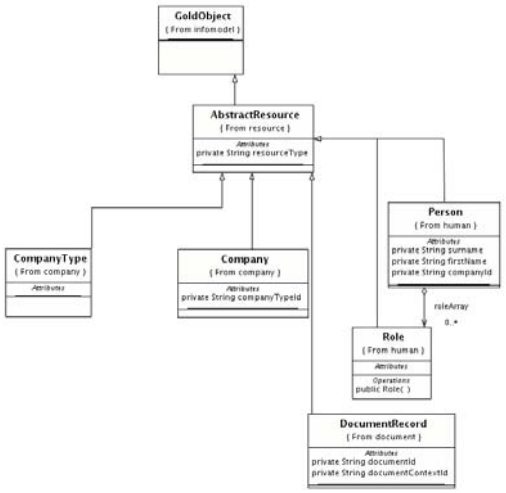


Figure 1: Gold Objects

The security within the GOLD Middleware is based around the `AbstractResource` class. Policies are defined which restrict access to resources based upon the `Roles` that the `Person` attempting to access the resource holds. These policies are based upon sets of rules that can be configured using a policy GUI and are stored as eXtensible Access Control Markup Language (XACML, OASIS, 2003) documents within the information repository.

Documents generated during the CPD process are stored within the `GoldDocument` class hierarchy illustrated below in Figure 2.

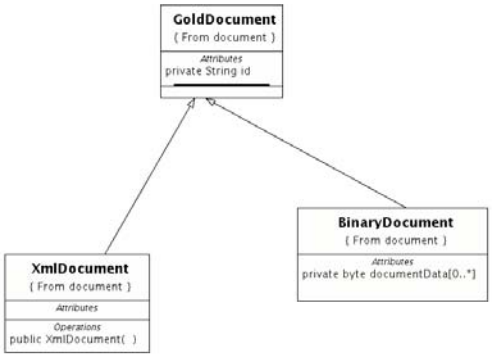


Figure 2: Document Class Hierarchy

The basic Information Model contains two document types: `XmlDocument`, which contains a single XML file and `BinaryDocument` which contains arbitrary binary data (such as, for example, a Microsoft Word document). Documents specific to the CPD process are derived from one of these two base classes. In some cases, such as the representation of chemical structures using the Chemical Markup Language (CML, Murray Rust, et al, 1995) XML schema exist for chemical specific information and can be easily stored within the information repository as `XmlDocument` objects. In other cases information is available in a structured form such as plant design data, however there is no standardised XML schema to represent this data. The GOLD project is actively investigating existing schema to represent this type of data, however, most information of this type is currently stored as binary data and persisted as `BinaryDocuments` within the information repository.

Documents within the Information Repository are referenced by means of `DocumentRecord` objects, which derive from `AbstractResource`, and contain details regarding the type of document data stored, access control restrictions, meta-data for searching etc. `DocumentRecord` objects are used to provide richer functionality than simple document identifiers and also to minimise the load on the application server when listing documents and performing other manipulations of documents that do not require the document data stored to be physically modified. The storage services or Information Repository, provided by the GOLD Middleware comprise the following physical web service:

Operation	***Description***
createDocument	Creates a new empty `DocumentRecord` in the information repository
retrieveDocument	Retrieves the document data associated with a specified `DocumentRecord`.
updateDocument	Updates the stored document data associated with a specified `DocumentRecord`.

Table 3: Storage Web Service

6. Security Services

The security services implemented within the GOLD Middleware depend largely on the specific roles held by VO participants. When applied to the CPD domain, these roles can be assumed to analogous to the skills described in Table 1.

Within the GOLD Middleware, security constraints are expressed and stored using XACML (OASIS, 2003). The implementation allows access control restrictions to be specified in terms of access permitted to VO resources based upon the roles that users hold.

The security services provided by the GOLD Middleware are designed to allow VO resource providers to authenticate users, identify the roles that users hold and to determine whether users should be permitted access to specified VO resources. This functionality has been implemented as two separate web services:

User authentication is provided by the Authentication Web Service, which contains a single method for verifying a username password pair. This functionality allows, for example, a custom JAAS (Java Authentication and Authorisation Service) module to be created and used within the Gridshpere portal server, which provides the end-user GUI, to authenticate portal users against the GOLD Information Model.

Operation	***Description***
authenticateUser	Authenticates a VO user with a username and password.

Table 4: Authentication Web Service

The facility to determine whether VO users can access certain resources is provided by the Authorisation Web Service. This wraps the XACML Policy Decision Point (PDP) into a web service that can provides methods that can be used by service providers within the VO to make access control decisions.

Operation	***Description***
readResource	Determines whether a VO user is permitted to read a specified resource.
writeResource	Determines whether a VO user is permitted to modify a specified resource.
performAction	Determines whether a VO user is permitted to perform an arbitrary action. This is possible as the XACML standard enables arbitrary actions to be specified as text strings.

Table 5: Authorisation Web Service

7. Co-Ordination Services

Co-ordination services within the GOLD Middleware are provided to ensure that actions performed by VO participants occur in the correct sequence and at the correct time to enable the work performed by the VO to proceed. Because of the highly dynamic nature of the CPD process, a flexible approach to co-ordination has been implemented in the current incarnation of the GOLD Middleware. The approach adopted has been to model VO projects as sets of discrete Tasks, each of which can contain a number of `DocumentRecords` corresponding to the dossiers or individual documents required in order to consider the Task instance complete. Each Task has a number of attributes such as start and end dates, description text, comments, role membership etc. Because VO projects and Tasks derive from the `AbstractResource` class, access control is possible through the standard XACML policy mechanisms provided by the Security Services.

By capturing projects as lists of tasks with start and end times, management and monitoring can be carried out using familiar Gantt charts. Co-

ordination functionality is provided to VO members via the Project Web Service which allows access to projects, tasks and their constituent documents.

Operation	*Description*
getTaskDocument	Get specified `DocumentRecord` associated with a specific project task
saveTaskDocument	Save a `DocumentRecord` to a specific project task.
getProject	Get a specified `Project` object
getTask	Get a task associated with a specified project.

Table 6: The Project Web Service

In addition to the task based approach for and co-ordinating projects on a high level, there is a need to initiate and control interactions between individual VO participants. For example, when a project requires the production of a specific document, a mechanism is needed to communicate that requirement to the relevant parties. In order to accommodate this requirement, the GOLD Middleware uses a "Worklist" approach whereby each VO User has a list of tasks to be performed. Events such as the beginning or end of tasks, the arrival of new VO members or the need for the production of project documents can be brought to the attention of selected VO users by placing an appropriate message into their Worklists. This has been implemented as a Messaging web service that can be used to send messages to individual VO users or to all users that are members of a specific VO role.

Operation	*Description*
sendUserMessage	Sends a message to the Worklist of a specific VO user.
sendRoleMessage	Sends a message to the Worklists of all users with a specified role.
getMessages	Retrieve all of the messages for a specified VO user.

Table 7: Messaging Web Service

8. Regulation Services

The Regulation Services provided by the GOLD Middleware are responsible for monitoring the interactions between individual users and companies so that actions performed within the VO comply with agreed standards of operation. There are numerous ways to define these standards: it could be a requirement that responses to requests for documents and information are returned within a pre-defined time interval. It may also be desirable that certain requests and interactions are non-repudiable such that no party can deny these interactions at a later date.

The GOLD Middleware has provided two mechanisms for performing regulation. The first is a logging service that accepts log messages from all of the other components of the Middleware, thereby allowing a thorough auditing (or possible replay) of the events and messages that were exchanged over the course of a VO project. The logging web service stores the log messages contained in the class hierarchy, a partial view of which is shown in Figure 3, in the Information Repository database.

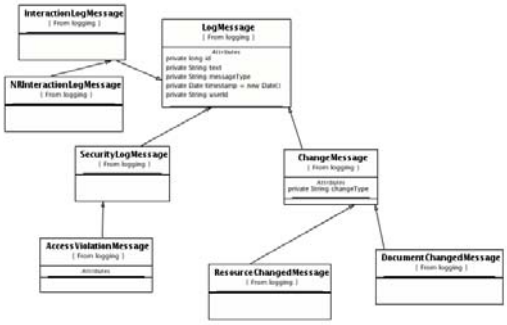


Figure 3: Logging Classes

The logging web service provides the following functionality to store and search for log messages (Table 8):

Operation	***Description***
logMessage	Sends a message to the logging service which is stored in the Information Repository database.
searchLog	Searches the Information Repository database for log messages of a certain type or that were logged within a certain time period or pertaining to a specific resource or user.

Table 8: Logging Web Service

The second regulation mechanism provided by the the GOLD Middleware uses the non-repudiable exchange tools developed by Cook, et al, 2002 to ensure that communications between individuals for certain classes of interaction are impossible to deny at a later date. For the sake of integration, the non-repudiation tools use the logging service to save information regarding the state of any non-repudiable information exchange between participants.

Whilst the non-repudiation protocol as implemented by Cook, et al, 2002, is complex, on a simple level, the non-repudiation system acts as a set of Web Service handlers that intercepd messages flowing between VO participants that need to be non-repudiable. The effective structure of this system is shown in Figure 4 which shows a message, **msg**, being transferred from participant **A** to participant **B** via a trusted Delivery Agent, **DA**.

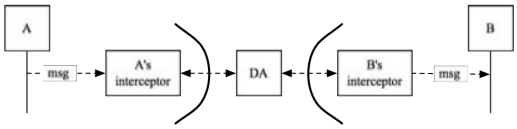


Figure 4: Non-Repudiable Message Delivery Structure

9. Implementation Details

The current incarnation of the GOLD Middleware has been implemented within the Sun Microsystems Application Server v9. This has been selected partly based upon its open source status and partly because it provides, in conjunction with Netbeans 5.0, an easy to use environment for developing and deploying Web Services. The majority of GOLD Middleware functionality has been implemented as Enterprise Java Beans (EJBs) with Web Service wrappers provided to deliver the services described above. In order to reduce development effort, data is exchanged between web services in the form of XML documents which are currently automatically mapped to Java classes using the XML Serialization provided by the Java runtime. Future work, however, will define formal XML schema for the representation of the class attributes within the Information Model and use Java XML Binding (JAXB) to exchange information in a more interoperable format. Storage of the data within the Information Repository has been implemented using the Hibernate object-to-relational database mapping library (http://www.hibernate.org), which automatically creates database schema and handles all serialisation / deserialisation and synchronisation issues. The database used for the demonstration system is the open source MySQL package, although use of the Hibernate library provides a high degree of database independence for a production quality system.

Configuration of the VO users, roles and policies is performed using a Java Server Pages (JSP) application, whilst the GUI presented to End Users is based upon the Gridshpere portal server, with additions to support authentication of users via the GOLD authentication web service. In order to demonstrate the potential for multiple views into an operating VO, the security policies are configured using a Swing GUI which interacts directly with the security EJBs using CORBA over SSL.

10.Conclusions

The Middleware developed as part of the GOLD project has been guided by a number of factors ranging from the experience of the individual team members operating actual CPS projects to a series of interviews performed with a significant number of CPD companies in conjunction with the Management School at Lancaster University. The current implementation, whilst able to support the CPD process has been designed with flexibility in mind. By modifying the security roles (Section 6) and document types supported (Section 5), collaborative development projects across a wide range of application domains can be supported. This generic approach is an important aspect of the GOLD project, as one of the key requirements is to support multiple classes of VO.

11.References

Demchenko, Y., 2004. Virtual organisations in computer grids and identity management. Information Security Technical Report, 9, 1, 59-76.

Cook, N., Shrivastava, S. and Wheather, S. 2002. Distributed Object Middleware to Support Dependable Information Sharing between Organisations. In Proceedings of IEEE Int. Conf. on Dependable Systems and Networks (DSN), Washington DC, USA.

Skonnard, A, 2002, The XML Files: The birth of Web Services, MSDN Magazine, Volume 17, Number 10, October 2002.

Sharman, N., Alpdemir, N., Ferris, J., Greenwood, M., Li, P. and Wroe, C. 2004. The myGrid Information Model. In Proceedings of the UK e-Science All Hands Meeting 2004, 1 September.

OASIS. 2003. eXtensible Access Control Markup Language (XACML) Version 1.0. OASIS Standard, http://www.oasis-open.org/committees/xacml.

Conlin, A., Cook, N., Hiden, H., Periorellis, P., Smith (2005) RCS-TR: 923 GOLD Architecture Document. School of Computing Science, University of Newcastle, Jul 2005

Murray-Rust, P., Rzepa, H.S., Leach, C., 1995, CML – Chemical Markup Language, Poster presentation, 210th ACS Meeting, Chicago, 21st August, 1995

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

VOMS deployment in GridPP and NGS

Alessandra Forti[1],

Mike Jones[2], Sergey Dolgobrodov[1]

[1)] School of Physics and Astronomy, [2)] Manchester Computing,
The University of Manchester, Manchester UK

Abstract

We describe our experience in practical deployment of the gLite VOMS (Virtual Organization Management Service). The formation of all sizes of groups with similar research agenda can be viewed as the emergence of Virtual Organisations (VOs). The deployment of centralised VOMS servers at a national level as part of a grid infrastructure facilitates the creation of these groups and their subsequent registration to the existing grid infrastructures. It also helps grid resources to engage user communities by dealing with them in a more scalable way. In the following we will describe the use cases, some of the technical aspects, and the deployment and administration of such a VOMS server. The evaluation of robustness of the VOMS releases 1.4, 1.5 and 3.0 and known problems are also described.

1. Introduction

In a grid environment there is a fundamental difficulty in the granting access rights to resources: users are no longer able to be recognised through their institutional login procedures nor do they belong to well defined local entities which are recognised outside those institutes.

Not long after the emergence of the Grid paradigm came the necessary concept of the Virtual Organisation (VO). VOs were created to solve the problem of identifying groups of abstract entities in online communities. These communities have, in some cases, become identified with groups of user. Access control policies combined with this type of VO have enabled grids to construct middleware to decide who gets access to what on their grid. VOs don't have to have geographical, administrative or even national boundaries. For this reason a strict definition of a VO is difficult to reach consensus upon.. For clarity here we will define a VO as a set of users who have signed an AUP (Acceptable Usage Policy) and that that VO in its own right is able to be given the authority to use a certain percentage of resources under different administrative control.

In the particle physics community for example it has long been understood that each High Energy Physics (HEP) experiment maps neatly onto a VO. Each experiment gets access to its resources and takes its share in the total number of cycles on the HEP's grid when available.

There are many other ways to form VOs besides this all of which would carry huge overheads if all VO members needed to register with all resources available in the emerging grids today.

2. Authorisation on a Grid and the choice VOMS

Authorisation in a grid context is the process which determines whether an entity may gain access to a resource within that grid. Usually by the time a resource is making an authorisation decision for an incoming request the authenticity of that request has been determined. This separation is key to the Public Key Infrastructures used in a number of grids today. It allows a user to identify themselves using a robust and universally acceptable token, and authorisation decisions to be made against that identity. This provides a useful separation aligning an identity (an entity will usually only have one of these) with an assertion from an authority which is well known, at the same time keeping that identity's properties and attributes (e.g. group membership, rolls, capabilities, etc.) separate from this assertion. Allowing a more scalable processes to later assert necessary properties by which authorisation can be determined.

It is for Authorisation that VOMS has been chosen as the provider of authorisation attributes in the GridPP and NGS grids. It provides all the modern security techniques like Single-sign on, delegation, non-repudiation and many more.

Thus, unlike most other security projects, VOMS does not focus on developing password based and local account based security applications or services. Instead, VOMS is meant to act as a server capable of securing all

services in a Grid environment, and exposing their functionality without putting resources and services in jeopardy. This enables the resource providers and grid enabled service provider to share and host their service and resource with full confidence which will help Computational Grid to grow.

3. VOs support in UK

3.1 Examples of local use cases

- Small site group with no resources.
- Small national experiment or research projects like MINOS, CEDAR and RealityGrid with no resources.
- A local group of a bigger VO that owns local resources and do not want to share them but want to access them with the same set of tools. (Typical of this case are data servers for local users.)
- Grid application development test machines.
- Different grids might want to cooperate to support each other resources.
- The local funding situation imposes to share resources with groups not belonging to any VO; however they might be willing to access their portion of resources through the grid.
- Distributed Tier 2 sites[1] might want to unify certain categories of users in one VO.
- A site might want to give temporary access to one group resources to another local group.

3.2 GridPP and NGS

In UK two different national grid organizations GridPP and NGS (National Grid Service) have decided jointly support gLite VOMS servers.

- Common infrastructure to maintain the VOMS servers
- Common VOs support
- Common distribution of information
- Enable each other VOs on each other systems

The pool consists of two front end servers one for NGS and one for GridPP, two backup servers and a test server. The servers are hosted at the in Manchester as part of the Tier2 and NGS infrastructures. They have been running since January 2006 and they now host 9 national VOs and 2 local.

[1]A tier 2 site provides regional Compute Power within the EGEE and GridPP.

3.3 Enabling a VO

A formal request has to be made to the management. The following information about the VO has to be supplied in the request.

- VO name that conforms to the LCG/EGEE guidelines. The latest format was DNS format to avoid names for different VOs clashing. For example minos.gridpp.ac.uk is an acceptable name. However short UNIX like user names are still used for practical reasons.
- VO support contacts. Specific people and mailing lists.
- Security contacts - two people who can respond quickly in the event of a security incident relating to a member of the VO, or to the VO as a whole.
- VO services end points like file catalogs.
- Hardware requirements - memory size, disk space etc.
- Software requirements - any software beyond the basic Linux tools/libraries, including things which are part of standard distributions as they may not be installed by default.
- Typical usage pattern - expected job frequency and variation over time, job length, data read and written per job etc.
- Glue schema fields used - this would give an idea of what is really used in the information system and needs to be ensured to be properly set and maintained.
- General procedures – like VO software installation
- Roughly the number of users expected to use the resources, to give a guide to how many pool accounts to create.

The request has to be approved by the GridPP/NGS management. After approval the VO gets created on the VOMS server and the VO manager is enabled to add users. Sometimes the VO is too small and the VO administration is done by the VOMS manager under request. The information to enable the VO at sites is then downloadable from the GridPP/NGS WEB sites. VOs are considered responsible for the maintenance of the information in their own interest.

4. Technical aspects of VOMS

4.1 gridmap-files to VOMS awareness; from individual authorization to virtual organisation

Production grid services providing simple access to data and compute resources have, to date, dealt with authorization on an individual by individual base. There has been a trend to supply some level of delegation to this process.

VOMS mechanisms have the potential to provide a level of delegation such that authorisation decisions may be taken based on membership to a virtual organization, removing the requirements of pre-registration of individuals. This section describes the evolution of these mechanisms.

4.2 gridmap-files

The purpose of a gridmap-file is to translate an incoming request whose originator's identity is known as a string representation of an X509 namespace (their distinguished name, DN) to a local system identity: the username of a local account. It is a flat file, each line containing a DN in double quotes followed by a comma separated list of local user identities. The service gateway having obtained the user's DN through the context of the SSL negotiation searches the gridmap file line by line until it finds a match. Both the account and the mapping must exist before the user is authorised to use the service.

4.3 Pool Accounts and LDAP directories

The prerequisite of a system account forces each prospective user to register with each system. In a grid comprising of many resources and many users this registration process and account generation is not scalable. The Pool account mechanism goes some way to providing a mechanism with which to address this. Pool accounts system simply leases system accounts to end grid users for a specified duration.

Having only partially addressed the registration issues (accounts are available but authorization to use them has still to be granted) a system of populating the gridmap file automatically is used. This system creates mappings to sets of (pools of) accounts. To date this has been achieved by regularly retrieving users' DNs which have been published by various organisations in secure LDAP servers. With all of the above in place access to resources can be granted to predefined communities; membership of these communities can be maintained by the communities themselves. At the time of writing, this is where authorization policies are realised in production grids like the UK National Grid Services. The authorization mechanisms as described above have a number of drawbacks:

.

On line dependence for maintenance of the gridmapfile,

- Denial of service attacks changes the behaviour of the communities membership,
- No ability to deal with users who are members of multiple communities (Group membership),
- No ability to select differing levels of access within a recognised communities (roles and capabilities),
- Data Protection.

Out of the LCG and its sister project gLite has emerged more elegant authorization mechanisms. While the concept has been around for a while stable implementations are still relatively new.

4.4 lcas and lcmaps

LCAS (Local Centre Authorization Service) and LCMAPS (Local Credential Mapping Service) replace the gridmap file system with something a little more sophisticated. As the names suggest they split the process into separate authorization and the mapping mechanisms. LCAS is a plug-in[2] which is called from within a modified service, (currently there exists modified versions of the Globus gatekeeper and an earlier version of the GridFTP server). LCAS makes Authorization decisions based upon three distinct sets of policy data: users/groups allowed, users/groups disallowed[3], and service availability times. LCMAPS (Local Credential Mapping Service) handles the assignment of local accounts and credentials. LCMAPS is the enforcement point of the authorization decision made by the LCAS.

4.5 VOMS Awareness

The VOMS provides trustable assertions relating group memberships, roles and capabilities to the owner of X509 certificates. It maintains a database of these relationships and a means to administer them. It also provides a web service through which these assertions can be obtained. These assertions can be obtained by an authorised resource or by the individual to whom the assertions belong.

In the First case, where the resource obtains these assertions and applies them, provides little more than today's pool account/LDAP system. The full advantages of the VOMS appears when

[2] A service implementation of LCAS is also being developed.

[3] Rather like the hosts.allow and hosts.deny file used by tcp wrappers.

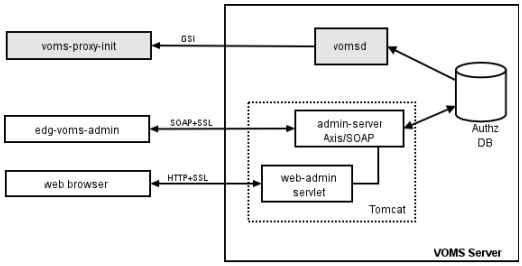
the user is able to download assertions and hand them over to the requested service, thus providing a mechanism by which the user can assert their choice of group membership, role and capability from those granted to them. It is this ability that fully addresses the five drawbacks listed above.

5. VOMS Deployment

5.1 The Virtual Organization Membership Service. Overview

The Virtual Organization Membership Service (VOMS) has been developed in the framework of EDG and DataTAG collaborations to solve the problems of granting users authorisation to access the resources at VO level, providing support for group membership, roles and capabilities. The VOMS system consists of the following parts

- User server: receives requests form client and returns information about the user.
- User client: contacts the server presenting a user's certificate and obtain a list of groups, roles and capabilities[4] of the user.
- Administration client: used by VO administrator to add user, create new groups, change roles.
- Administration server: accept the request from the admin client and updates the database.



GridPP and NGS Grid environments in Manchester University

5.2 gLite VOMS (MySQL)

The recent release version 3.0 based on MySQL database is rolled out with the EGEE gLite 3.0 Middleware in Manchester HEP group to support several local VOs. The URI for the GridPP VOMS service is https://voms.gridpp.ac.uk . This contains an alias name for two Intel(R) Pentium(R) 4 CPU 3.20GHz with 1GB RAM and 160GB HDD. The one is front-end voms01.gridpp.ac.uk and the second one is backup voms02.gridpp.ac.uk. This name scheme allow to keep service running by moving main alias name to the backup server in case of any serious problems with the main one. The hosts are running under Linux SL3, kernel 2.4.21. Both hosts are now in the "production stage".

Currently there are 10 VOs defined on the GridPP server:

- gridpp – for the GRIDPP project, higher energy physics community
- ltwo – Teaching purposes within the London Tier 2
- t2k – Next Generation Long Baseline Neutrino Oscillation Experiment
- minos – Main Injector Neutrino Oscillation Search
- cedar – Combined e-Science Data Analysis Resource for high-energy physics
- gridcc – for the GRID project on remote control and monitoring of instrumentation such as distributed
- manmace – for the Manchester MACE engineers to run on the resources maintained at the Manchester EGEE Tier2 centre
- babar – for running babar experiment simulation and analysis on the EGEE/LCG grid
- pheno – dedicated to developing the phenomenological tools necessary to interpret the events produced by the LHC
- ralpp – VO for local tests in the RAL Particle Physics department

Another VO for Mathematicians in ScotGrid Tier2 is under process of approval.

NGS has chosen to support a global NGS VO for the moment.

The data reside on the same host in MySQL data base (version 4.11 server and client). There over 25 entries across the VOs in this data base now.

The machine has been very stable; it presents today an uptime over 120 days with one occasional reboot due to exceeded the maximum of process threads, although this may be explained by a modest number of entries and queries. The average number of connections for each VO was 1800 per day.

A similar set up hosts the NGS VOMS server. In this case there is currently one group, that describing the current list of NGS members.

Apart of exploiting version 3.0 we did run the 1.4 and 1.5 releases in the past and found the new version much more stable, many problems have been fixed.

[4]The Capability attribute although still valid in the VOMS attribute certificates is deprecated and no longer able to be produced through the VOMS interfaces

5.3 Known problems

There was a difficult situation with VOMS of versions 1.4/1.5 development and support. Bug samples:

1. VOMS -admin (on MySQL) doesn't list users with more than one Role (https://savannah.cern.ch/bugs/index.php?func=detailitem&item_id=14398);
2. Move the data base from older version of the VOMS based on MySQL to a newer one (e.g. from 1.3 to 1.4 or from 1.4 to 1.5), leads to hangs the VOMS and VOMS-admin tool. Sometimes it happens because of different database frame for different versions of VOMS;
3. Bad tomcat performance affecting VOMS access and gridmap file generation (https://savannah.cern.ch/bugs/?func=detailitem&item_id=14057).
4. VOMS-admin hangs due to tomcat "OutOfMemoryError" (https://savannah.cern.ch/bugs/?func=detailitem&item_id=16250).

The last problem with the tomcat is still presented in version 3.0 and has to be addressed in a new release.

6. Discussion

Our experience in deployment of gLite VOMS service shows that the current state of the middleware is acceptable to serve our needs. An increasing number of VO creation requests are being submitted to satisfy the most diverse necessities of local and regional groups.

7. Acknowledgemnts

This work was funded by the Particle Physics and Astronomy Research Council through their GridPP end e-Science programmes and by NGS. We would also like to thank other members of the EGEE security working group for providing much of the wider environment into which this work fits.

References

1. gLite VOMS Core User and Reference Guide: *https://edms.cern.ch/file/571991/1/voms-guide.pdf*
2. gLite VOMS Admin Tools User and Reference Guide: *https://edms.cern.ch/file/572406/1/user-guide.pdf*

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

The RealityGrid PDA and Smartphone Clients: Developing effective handheld user interfaces for e-Science

Ian R. Holmes and Roy S. Kalawsky

East Midlands e-Science Centre of Excellence,
Research School of Systems Engineering, Loughborough University, UK

Abstract

In this paper the authors present the RealityGrid PDA and Smartphone Clients: two novel and highly effective handheld user interface applications, which have been created purposefully to deliver (as a convenience tool) flexible and readily available 'around the clock' user access to scientific applications running on the Grid. A discussion is provided detailing the individual user interaction capabilities of these two handheld Clients, which include real-time computational steering and interactive 3D visualisation. The authors also identify, along with several lessons learned through the development of these Clients, how such seemingly advanced user-oriented e-Science facilities (in particular interactive visualisation) have been efficiently engineered irrespective of the targeted thin-device deployment platform's inherent hardware (and software) limitations.

1. Introduction

The UK e-Science RealityGrid project [1] has extended the concept of a Virtual Reality (VR) research centre across the Grid. "RealityGrid uses Grid technologies to model and simulate very large or complex condensed matter structures and so address challenging scientific problems that would otherwise remain out of reach or even impossible to study" [2]. As part of the project's ongoing research initiative, the authors (in part) conducted a very thorough and comprehensive Human Factors evaluation [3]; finely examining e-Science user interaction methodologies with a view towards improving everyday working practices for applications (as well as computer) scientists. The result and subsequent evaluation of this investigative process highlighted (amongst other notable findings) the need for a more flexible and convenient means of supporting ubiquitous scientific user interaction in Grid computing environments. This need was discerned from RealityGrid's scientific workflows in which it was identified that at certain stages researchers could benefit from having a more lightweight and portable extension/version of their familiar desk-based front-end applications; also (in some cases) providing a more attractive and usable alternative to some of today's current and otherwise somewhat cumbersome command line-driven systems. Creating a more convenient and mobile form of e-Science user interaction was deemed beneficial because it would allow researchers the freedom to interact with their applications from virtually any location and at any time of day (or night); especially supporting (and sustaining) the vitally important interactive process at times when it may become inconvenient or otherwise impractical for the scientist to carry a laptop PC or to connect to the Grid from an office, laboratory or other similar statically-oriented working environment.

In this paper the authors describe how this discerned need for convenient e-Science usability has been addressed (within the RealityGrid project) through the creation of the RealityGrid PDA and Smartphone Clients: two novel and highly effective handheld user interface applications, which have been built purposefully to deliver (as a convenience tool) flexible and readily available 'around the clock' user access to scientific applications running on the Grid. A discussion is provided detailing the individual user interaction capabilities of these two handheld Clients, which include real-time computational steering and interactive 3D visualisation. The authors also identify, along with several lessons learned through the development of these Clients, how such seemingly advanced user-oriented e-Science facilities (in particular interactive visualisation) have been efficiently engineered irrespective of the targeted thin-device deployment platform's inherent hardware (and software) limitations.

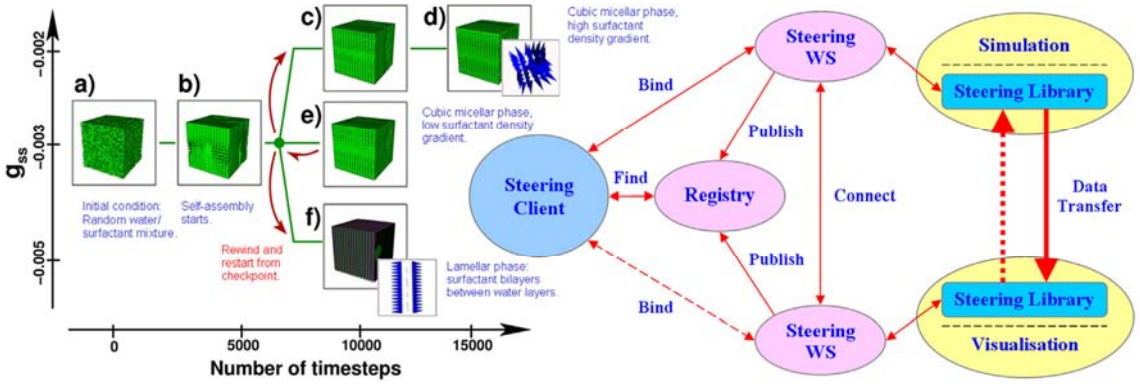


Figure 1. "The central theme of RealityGrid is computational steering" [2] (left). The project has developed its own Grid-enabled computational steering toolkit [10], comprising of lightweight Web Services middleware and a programmable application library (right).

2. Related Work: The UK e-Science RealityGrid Project

"Through the linking of massive computational infrastructure at remote supercomputing centres as well as experimental facilities, RealityGrid is providing scientists with access to enormous resources in a highly efficient manner, which means researchers in a range of fields can benefit from the close coupling of high-throughput experimentation, modelling and simulation, and visualisation" [4]. To give an insight into the computational scale of the project: RealityGrid has been central to the development, deployment and testing of scientific Grid middleware and applications, which, as demonstrated by the TeraGyroid [5], STIMD [6] and SPICE [7] projects, efficiently utilise a federated Grid environment (or 'Grid-of-Grids') built around the UK's most advanced computing technology and infrastructure, linked via trans-Atlantic fibre to the supercomputing resources of the US's TeraGrid [8].

2.1 Computational Steering

"The central theme of RealityGrid is computational steering, which enables the scientist to choreograph a complex sequence of operations or 'workflows'" [2] (Refer to Figure 1). "Using the Grid and a sophisticated approach called computational steering, the scientists steer their simulations in real-time so that the heavy-duty computing can focus where the dynamics are most interesting" [9]. RealityGrid has developed (as part of a wider Grid-based problem-solving environment) its own dedicated computational steering toolkit [10], comprising of lightweight Web Services middleware and a programmable application library (Refer to Figure 1). Using the toolkit, researchers have been able to inject the project's computational steering facilities into their existing (and traditionally non-interactive) scientific codes and thus become empowered to steer their experiments (simulations) in real-time on (or off) the Grid.

2.2 Lightweight Visualisation

Extended RealityGrid research has recently led to the creation of Lightweight Visualisation (Refer to Figure 2). Developed separately by the authors, Lightweight Visualisation has provided a lightweight software platform to support ubiquitous, remote user interaction with high-end scientific visualisations; as a standalone system it has also been designed for universal use both within and outside of RealityGrid. Through a framework of linked Grid-enabled components, Lightweight Visualisation provides the underlying software infrastructure to facilitate high-level visualisation-oriented user interaction via a wide range of inexpensive and readily available commodity user interface devices, such as PDAs (Refer to Figure 2), mobile phones and also low-end laptop/desktop PCs; systems all commonly affiliated by their inherent lack of specialised (expensive) hardware resources, which would traditionally have discounted them from being considered as viable user interaction outlets for computationally-intensive visualisation. By also adopting a similar architectural design to that of the established RealityGrid computational steering toolkit (Refer to Figures 1 and 2), Lightweight Visualisation services (as with steering) can be easily injected into existing visualisation codes, thus allowing researchers to interact remotely (via a PDA, for example) with dedicated user-oriented facilities, which are indigenous and often unique to their familiar or preferred visualisation tools/applications.

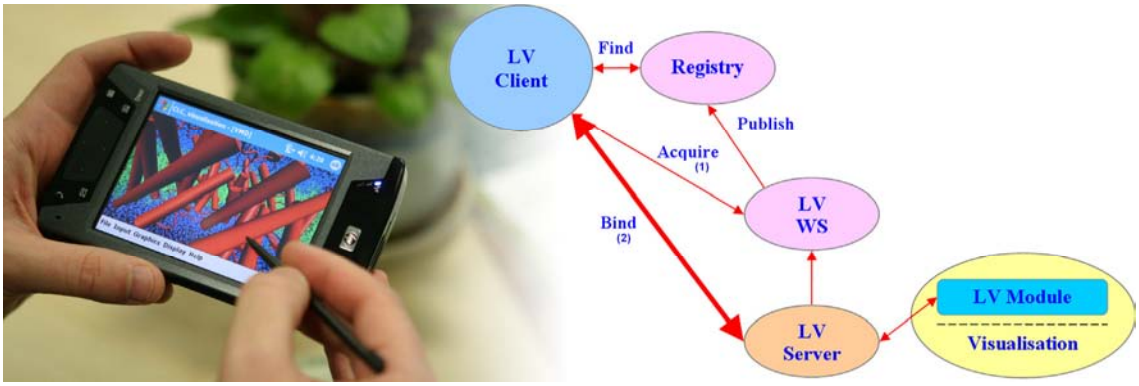


Figure 2. Lightweight Visualisation (right) has provided a lightweight software framework to support remote and ubiquitous visualisation-oriented user interaction via a wide range of inexpensive and readily available commodity devices such as PDAs (left), mobile phones and also low-end laptop/desktop PCs.

3. The RealityGrid PDA and Smartphone Clients

The RealityGrid PDA and Smartphone Clients have both been created to allow scientists the mobile freedom to stay in touch with their experiments 'on the move' and especially at times when it isn't always convenient (or practical) to carry a laptop PC or to connect to the Grid from a desk-based terminal in an office or laboratory-type environment. Using either of these two handheld e-Science Clients the researcher becomes able to readily access and freely interact with their scientific applications (discreetly if necessary) from virtually any location and at any time of the day (or night); connecting to the Grid over a standard, low-bandwidth Wi-Fi (802.11) or GSM/GPRS/3G network and interacting through a user device, small and compact enough to be easily transported around 'hands free' inside one's pocket.

The RealityGrid PDA and Smartphone Clients have each been individually engineered for their respective pocket-sized devices using C# and the Microsoft .NET Compact Framework; thus allowing them to be deployed onto any PDA or Smartphone device running on the Microsoft Windows Mobile embedded operating system. These two handheld e-Science user interface applications are each able to offer a comprehensive assemblage of high-level RealityGrid-oriented user interaction services, collectively allowing researchers to: find their scientific jobs running on the Grid (Refer to Figure 3), computationally steer their simulations in real-time (Refer to Figure 4), graphically plot parameter value trends (Refer to Figure 4) as well as explore on-line scientific visualisations (Refer to Figures 5 and 6).

3.1 Middleware Interactivity

The process by which the handheld RealityGrid Clients communicate with scientific simulation and visualisation applications, running on (or off) the Grid, is achieved via a set of bespoke proxy classes (internal to the Clients), which have been developed specifically to support interactivity with RealityGrid's lightweight middleware environment. Earlier versions of the RealityGrid middleware were built around the Open Grid Services Infrastructure (OGSI) specification and implemented using OGSI::Lite [11] (Perl). In this instance it was possible to automatically generate the Clients' internal communication proxy code using a Microsoft tool (wsdl.exe), which was fed with each individual middleware component's descriptive Web Services Description Language (WSDL) document. For the later versions of RealityGrid middleware, which have been built around the newer Web Services Resource Framework (WSRF) specifications and implemented using WSRF::Lite [11] (again in Perl), the Clients' internal proxies had to be programmed manually using bespoke low-level networking code. This was due to the fact that the proxy generation tool could not cope with the latest WSRF specifications and essentially failed to generate any usable code. Both of the handheld Clients (PDA and Smartphone) are each able to operate transparently with the earlier OGSI-based versions of RealityGrid's middleware as well as the latest WSRF-based implementation.

3.2 Security Provision

Through their respective internal proxy classes, the PDA and Smartphone Clients are both able to consume RealityGrid Web/Grid Services via a two-way exchange of Simple Object Access

Figure 3. The RealityGrid PDA and Smartphone Clients both provide a mobile facility to query a Registry (left) and retrieve a list of currently deployed jobs (right). Once a job has been selected, each Client will then attach (connect) remotely to the appropriate scientific application running either on or off the Grid.

Protocol (SOAP) messages, which are transmitted over a wireless network using the Hyper Text Transfer Protocol (HTTP) or (in the case of the latest WSRF-based middleware) with optional Secure Sockets Layer (SSL) encryption (HTTPS). In the event of using SSL, both Clients will require the user to specify a digital e-Science X.509 user certificate (or chain of certificates) in order to perform the Public Key Infrastructure (PKI) cryptography and thus encrypt/decrypt all of the data on the network. RealityGrid middleware (WSRF) will also specify a list of authorised users and will require a valid username and optional password to be inputted through the Client, which will then be inserted into a securely hashed Web Services Security (WSSE) header, inside every individually despatched SOAP packet. The .NET Compact Framework, which was used to build both of the handheld RealityGrid Clients, provided no built-in provision for the required WSSE or SSL/PKI security functionality. As a result these essential elements had to be custom-developed for both of the Clients using entirely bespoke C# code to implement WSSE and a relatively inexpensive third-party Application Programming Interface (API), known as SecureBlackBox [12], to implement SSL/PKI. Additional wireless (802.11) security can also be attained through the optional employment of Wired Equivalent Privacy (WEP) or Wi-Fi Protected Access (WPA/WPA2), both of which can be used in conjunction with the Clients' own security provision (when using Wi-Fi) but are not relied upon and therefore not discussed in any detail within the scope of this paper.

3.3 Finding Jobs on the Grid

When employing either of the RealityGrid PDA or Smartphone Clients, the first task for the user to perform (through the Client), prior to being able to steer a simulation or explore a visualisation, is to find their scientific jobs (the location of which may not be known) on the Grid. In RealityGrid, information describing each actively deployed job (its start date/time, creator ID, software package and most importantly its location) is published to a central (optionally secure) Web/Grid Service known as the Registry (Refer to Figures 1 and 2). The PDA and Smartphone Clients each provide the user with a convenient facility to query a RealityGrid Registry (of which there may be several) in order to retrieve a list of currently deployed (registered) jobs on the Grid (Refer to Figure 3). The process of requesting, retrieving and displaying job data is performed by both of the Clients with an average latency of roughly one quarter of a second; with a slightly longer additional time delay incurred in the event of having to first perform an SSL handshake with a Registry, prior to requesting its job data. Once a list of jobs has been successfully retrieved and displayed, the researcher is able to then select specific entries (jobs) to interact with on an individual basis. The PDA or Smartphone Client will then attempt to attach (connect) through the RealityGrid middleware to the appropriate scientific application (running either on or off the Grid), in turn unlocking the more advanced steering and visualisation-oriented features of the handheld user interface software.

3.4 Real-Time Computational Steering

Once successfully attached to a running simulation (having discovered its location from a Registry) the PDA or Smartphone Client will automatically display its built-in computational steering user interface (Refer to Figure 4). Through their respective steering interfaces, each of the handheld RealityGrid Clients provides the researcher with an effective mobile

Figure 4. The PDA and Smartphone Clients provide a mobile user interface for monitoring and tweaking (steering) simulation parameters in real-time (left), additionally incorporating visual parameter value trend plotting capabilities (right) to aid understanding of how experiments change and evolve as they run.

facility to monitor and tweak (steer) simulation parameters in real-time (either individually or as part of a batch), as the simulation itself runs and evolves concurrently on a remote computational resource. In RealityGrid, individual parameters are exposed from within a scientific application through the project's open-source computational steering API. This level of exposure facilitates internal simulation/application parameter data to be accessed and manipulated (steered) remotely via the RealityGrid PDA and Smartphone Clients, through a representative middleware interface known as the Steering Web Service (SWS) (Refer to Figure 1); or in the case of the earlier OGSI-based middleware: the Steering Grid Service (SGS). The RealityGrid PDA and Smartphone Clients, via remote interactions through the SWS/SGS, are both additionally able to provide mobile facilities for invoking lower-level simulation control procedures such as Pause/Resume and Stop/Restart.

Whilst attached to a simulation, the PDA and Smartphone Clients will each poll its representative SWS/SGS via SOAP message exchanges at regular one second intervals, to request and subsequently display 'up to the second' simulation parameter state information; with one integral parameter update (request and display) requiring a duration of roughly two tenths of a second in order to fully complete. The information returned to the Clients from the RealityGrid middleware (in SOAP format) individually describes all of the parameters that have been exposed from a simulation using the steering toolkit. Each parameter description includes a unique identifier (or handle), current, minimum and maximum values, data types, and status flags denoting 'steerable', monitored-only or internal to the application-embedded steering library. In order to maximise usability during the steering process, both user interfaces have been developed to utilise a sophisticated handheld device multi-threading scheme. This has enabled each of the Clients to continuously poll the steering middleware (for updated parameter information) in an integral and dedicated background thread, whilst concurrently remaining active and responsive at all times to capture user input events (entering/tweaking parameter values, etc.) in a completely separate user interface thread. Prior to the implementation of this handheld device multi-threading scheme, both Clients would simply 'lock up' their user interfaces and temporarily block all user inputs throughout the duration of every regular update polling cycle.

3.5 Visual Graph Plotting

To enhance and extend the process of steering a simulation through either of the handheld RealityGrid Clients, both user interfaces also incorporate a visual parameter value trend plotting facility (Refer to Figure 4). Using a plotted graph to visually represent parameter value fluctuation trends was identified from the initial Human Factors investigation [3] to greatly improve the scientist's perception and understanding of their experiments. The two handheld RealityGrid Clients will each allow the researcher to visually plot historical records for one user-specified parameter at a time. This visual plotting functionality has been custom-built (for both Clients) using the .NET Compact Framework's standard Graphics Device Interface API (known as GDI+). As with computational steering, visual graph plotting for both Clients is also performed in real-time, allowing the researcher to visually observe how their simulation changes and evolves concurrently, either naturally over a period of time or more often as a direct response to user-instigated computational steering activity.

Figure 5. The handheld RealityGrid Clients each offer the researcher an inclusive provision of advanced, visualisation-oriented interaction facilities, which (up until this point) would only typically otherwise have been accessible through conventional desktop (or laptop)-based user interface applications.

3.6 Handheld Interactive 3D Visualisation

The most advanced feature of the RealityGrid PDA and Smartphone Clients is their highly novel ability to each support user-interactive 3D visualisation (Refer to Figures 5 and 6). Both of the handheld RealityGrid Clients have been developed specifically for use with the authors' Lightweight Visualisation framework, allowing them to each offer researchers an inclusive provision of advanced, visualisation-oriented interaction facilities, which (up until this point) would only typically otherwise have been accessible through conventional desktop (or laptop)-based user interface applications. Using Lightweight Visualisation's middleware and it's accompanying, application-embedded module (Refer to Figure 2), the handheld RealityGrid Clients are both able to communicate user-captured interaction commands (pan, zoom, rotate, etc.) directly into a remote visualisation application, and in response receive visual feedback through the form of pre-rendered, scaled, encoded and streamed image (frame) sequences; processed and displayed locally within the Clients (using GDI+) at an average frequency of 7-10 frames-per-second. Thus Lightweight Visualisation is able to achieve an advanced level of 3D visualisation interactivity within each of the Clients by shielding their respective resource-constrained handheld devices from having to perform any graphically-demanding operations (modelling, rendering, etc.) at the local user interface hardware level.

The process by which the handheld RealityGrid Clients connect and interact remotely with a visualisation (via Lightweight Visualisation) is slightly more involved than for steering a simulation through the SWS/SGS. Having found and subsequently selected a visualisation application/job from a Registry (Refer to Figure 3), the researcher's PDA or Smartphone Client will then initiate a two-stage attaching sequence (Refer to Figure 2): firstly acquiring a remote host network endpoint from the Lightweight Visualisation Web Service; secondly establishing a direct socket channel (using the acquired addressing information) to the visualisation's representative Lightweight Visualisation Server. The Server is responsible for managing simultaneous Client connections, communicating received interaction commands directly into the visualisation application and encoding/serving pre-rendered images back to the Client(s) for display. All communication between the Client and the Server (message passing/image serving) is performed using Transmission Control Protocol/Internet Protocol (TCP/IP) with optional SSL data protection. The use of TCP/IP as the base networking protocol within Lightweight Visualisation was deemed preferable to the higher-level HTTP, in order to eliminate undesirable transmission and processing latencies incurred through the marshalling of large quantities of binary image data within SOAP-formatted message packets.

Once attached to a visualisation, user interactions through the handheld Clients adopt similar, standard conventions to those of the familiar desktop model: dragging the stylus across the screen (as opposed to the mouse) or pressing the directional keypad (as opposed to the keyboard cursor keys) in order to explore a visualised model, and selecting from a series of defined menu options and dialog windows in order to access any application-specific features of the remote visualisation software (Refer to Figure 6). As with steering, handheld device multi-threading has again been implemented within the Clients' 3D visualisation front-ends, enabling them to both continuously receive and display served image streams in a dedicated

Figure 6. When using the RealityGrid PDA or Smartphone Clients (left), researchers who employ VMD [13] on a daily basis are able to access and interact (through their handheld device) with familiar, indigenous user interface facilities from their everyday desktop or workstation application (right).

background thread, whilst constantly remaining active and responsive to capture all user input activity in a separate and concurrently running user interface thread.

3.7 Tailor-Made User Interaction

RealityGrid has delivered Grid-enabling support for a wide range of existing and heterogeneous scientific visualisation codes. These currently include commercial and open-source codes that have been successfully instrumented (adapted) for RealityGrid using the toolkit, as well as purely bespoke codes that have been written within the project to fulfil a specific scientific role; usually acting as an on-line visualisation counterpart to a bespoke simulation application. Each of RealityGrid's numerous visualisation applications typically provides its own unique assemblage of dedicated user interaction services, which are also often geared specifically towards a particular, individual field of scientific research (fluid dynamics, molecular dynamics, etc.). At the heart of Lightweight Visualisation is its ability to accommodate this level of user-interactive diversity by 'tapping into' any given visualisation's indigenous user facilities (through a small embedded code module or script) (Refer to Figure 2) and then exposing them (similar to steerable parameters) so that they can be accessed and invoked remotely by a specially built (or Lightweight Visualisation-enabled) Client; in this instance the RealityGrid PDA and Smartphone Clients.

To properly demonstrate the benefits of this 'tapping into' approach, the RealityGrid PDA and Smartphone Clients have both initially been developed to provide tailor-made handheld front-ends for the Visual Molecular Dynamics (VMD) [13] application (Refer to Figures 5 and 6). VMD is an open-source visualisation code that has been employed extensively within RealityGrid projects such as STIMD [6] and SPICE [7]; it has also provided the initial test-bed application for developing the Lightweight Visualisation framework. When using the 3D visualisation user interfaces of the RealityGrid PDA or Smartphone Clients, researchers who employ the VMD code on a daily basis are able to access and interact (through their handheld device) with familiar, indigenous user interface facilities (in addition to the generic pan, zoom, rotate, etc.) from their everyday desktop or workstation application (Refer to Figure 6).

4. Conclusion and Future Directions

In this paper the authors have presented the RealityGrid PDA and Smartphone Clients; offering an insight into how these types of handheld user interface applications (despite their perceived inherent limitations, both in terms of hardware and software capability) can now be effectively engineered to deliver real, beneficial scientific usability (steering, 3D visualisation, security, etc.) and provide flexible 'around the clock' user access to the Grid by complimenting and extending the more traditional desk-based methods of e-Science user interaction. Through the creation of these two handheld user interfaces, scientists within the RealityGrid project have been able to take advantage of a highly convenient means for readily accessing and interacting with their Grid-based research experiments, which has proved to be highly beneficial, particularly when having to deal with scientific jobs (often due to delayed scheduling allocations or lengthy compute-time periods) outside of working hours or whilst being away from the conventional desk/office-based working environment: during a meeting/conference, or whilst being at home in the evenings/away at weekends, etc.

Future avenues of development for the RealityGrid PDA and Smartphone Clients currently include creating additional tailor-made handheld front-ends for alternative visualisation codes (and scientists who don't use VMD). Extended development is also currently engaged with integrating a handheld user interface for RealityGrid's recently released Application Hosting Environment (AHE) [14], which will introduce invaluable new resource selection, application launching and job management capabilities. Further endeavours (outside of RealityGrid) are also currently geared towards deploying this developed handheld technology as part of a trial with the University Hospitals of Leicester NHS Trust (UHL NHS Trust), in which it will be employed by surgeons and medical consultants to view and interact with Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) scan data, whilst being away from the conventional office or desk-based working environment.

Acknowledgement

The authors wish to thank the dedicated teams of scientists within the RealityGrid project (University College London, University of Manchester, University of Oxford, Loughborough University, Edinburgh Parallel Computing Centre and Imperial College). Special thanks also go to Simon Nee (formally of Loughborough University) and to Andrew Porter of Manchester Computing for their invaluable, additional support and advise.

References

[1]. RealityGrid, http://www.realitygrid.org

[2]. J. Redfearn. Ed. (2005, Sept). "RealityGrid – Real science on computational Grids." *e-Science 2005: Science on the Grid*. [On-line]. pp. 12-13. Available: http://www.epsrc.ac.uk/CMSWeb/Downloads/Other/RealityGrid2005.pdf

[3]. R.S. Kalawsky and S.P. Nee. (2004, Feb.). "e-Science RealityGrid Human Factors Audit – Requirements and Context Analysis." [On-line]. Available: http://www.avrrc.lboro.ac.uk/hfauditRealityGrid.pdf

[4]. D. Bradley. (2004, Sept). "RealityGrid – Real Science on computational Grids." *e-Science 2004: The Working Grid.* [On-line]. pp. 12-13. Available: http://www.epsrc.ac.uk/CMSWeb/Downloads/Other/RealityGrid2004.pdf

[5]. R. J. Blake, P. V. Coveney, P. Clarke and S. M. Pickles, "The TeraGyroid Experiment – Supercomputing 2003." *Scientific Programming*, vol. 13, no. 1, pp. 1-17, 2005.

[6]. P.W. Fowler, S. Jha and P.V. Coveney. "Grid-based steered thermodynamic integration accelerates the calculation of binding free energies." *Philosophical Transactions of the Royal Society A*, vol. 363, no. 1833, pp. 1999-2015, August 15, 2005.

[7]. S. Jha, P. Coveney and M. Harvey. (2006, June). "SPICE: Simulated Pore Interactive Computing Environment – Using Federated Grids for "Grand Challenge" Biomolecular Simulations," presented at the 21st Int. Supercomputer Conf, Dresden, Germany, 2006. [On-line]. Available: http://www.realitygrid.org/publications/spice_isc06.pdf

[8]. TeraGrid, http://www.teragrid.org

[9]. M. Schneider. (2004, April). "Ketchup on the Grid with Joysticks." *Projects in Scientific Computing Annual Research Report*, Pittsburgh Supercomputing Center. [On-line]. pp. 36-39. Available: http://www.psc.edu/science/2004/teragyroid/ketchup_on_the_grid_with_joysticks.pdf

[10]. S.M. Pickles, R. Haines, R.L. Pinning and A.R. Porter. "A practical toolkit for computational steering." *Philosophical Transactions of the Royal Society A*, vol. 363, no. 1833, pp. 1843-1853, August 15, 2005.

[11]. WSRF::Lite/OGSI::Lite, http://www.sve.man.ac.uk/Research/AtoZ/ILCT

[12]. SecureBlackBox, http://www.eldos.com/sbb

[13]. W. Humphrey, A Dalke and K. Schulten, "VMD – Visual Molecular Dynamics", *J. Molec Graphics*, vol. 14, pp. 33-38, 1996.

[14]. P.V. Coveney, S.K. Sadiq, R. Saksena, S.J. Zasada, M. Mc Keown and S. Pickles, "The Application Hosting Environment: Lightweight Middleware for Grid Based Computational Science," presented at TeraGrid '06, Indianapolis, USA, 2006. [On-Line]. Available: http://www.realitygrid.org/publications/tg2006_ahe_paper.pdf

Ian R. Holmes and Roy S. Kalawsky 2006

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Application Reuse through Portal Frameworks

Mark Baker and **Rahim Lakhoo**

ACET
University of Reading

Distributed Systems Group
University of Portsmouth

{Mark.Baker@computer.org, Rahim.Lakhoo@port.ac.uk}

Abstract

Educational institutions, enterprises and industry are increasingly adopting portal frameworks, as gateways and the means to interact with their customers. The dynamic components that make up a portal, known as portlets, either use proprietary interfaces to interact with container that controls them, or more commonly today, the standardised JSR-168 interface. Currently, however, portal frameworks impose a number of constraints and limitations on the integration of legacy applications. This typically means that significant effort is needed to embed an application into a portal, and results in the need to fork the source tree of the application for the purposes of integration. This paper reports on an investigation that is studying the means to utilise existing Web applications through portlets via bridging technologies. We discuss the capabilities of these technologies and detail our work with the PHP-JavaBridge and PortletBridge-portlet, while demonstrating how they can be used with a number of standard PHP applications.

1. Introduction

Portal and portlet technologies are becoming increasingly popular with Web developers and orgnisations alike. Portals aim to provide a suite of applications in a common and customizable environment. Portal containers provide a number of portlet applications, which users can subscribe to or select. A portal would typically allow a user to organise a collection of portlets to their liking. Portlets themselves can be thought as mini Web applications, with GUI capabilities, such as being able to resize its window.

Portlets were once developed using vendor specific APIs and container, such as IBM's WebSphere. This resulted in portlets, which were confined to a single container. The JSR-168 [1] specification has had an impact on portal developers and their communities. Developers following the JSR-168 are now able to produce a single portlet application that is deployable across different portlet containers. However, developers are currently forced to either redesign or re-implement their existing applications with Java, which may be a deterrent for organisations to employ portlet technologies, especially if a viable Web enabled solution is already available.

Typically, scripting languages are used for Web development, which is a popular and well-established method of generating an application. Although portlets can use JavaServer Pages (JSP) to provide a GUI; this is not as widely used as other languages, such as PHP [2], Perl, Python and more recently Ruby. The number of tools and applications available to scripting languages is large compared to solutions offered based on JSP. Script based languages offer rapid development with low learning curves. Such an example is Ruby on Rails [3], which is a Web framework optimised for developer productivity.

More recently, developments have started to move towards bridging the gap between script based languages and Java. JSR-223 [4] aims to allow scripting language pages to be used in Java Server-side applications. JSR-223 is using PHP as their example scripting language. JSR-223 is not limited Web development or PHP. Groovy, JavaScript and command line interaction is currently also available. Currently JSR-223 is being distributed as part of Sun's JDK 1.6 "Mustang" beta. Other developments include the PHP-JavaBridge [5], which incorporates JSR-223, but is described as an XML-based network protocol, which can be used to connect to native script engines. PHP-JavaBridge features the ability for PHP applications to be executed within a J2EE container, such as Apache Tomcat. The PHP-JavaBridge will be discussed further in Section 2.2.

While JSR-223 and the PHP-JavaBridge project provide a means for Java to access script based languages, portal developers are also producing bridges. Apache Portals is developing Portals

Bridges [6], which will offer JSR-168 complaint portlet development using Web frameworks, such as Struts, JSF, PHP, Perl and Velocity. Bridges are being developed for each of the supported Web frameworks. In Portals Struts Bridge [8] support has been added to a number of portals, including JetSpeed, JBoss, GridSphere [9], Vignette Application Portal and Apache Cocoon. Portlet developers are also producing portlets, which are intended to allow regular Web sites to be hosted as JSR-168 compliant portlets. The PortletBridge-portlet [10] aims to proxy and rewrite content from a downstream Web site into a portlet.

The question is though, with these bridging technologies becoming more usable, can developers use them efficiently and avoid re-implementing existing applications within compliant portlets?

This paper will discuss the various implementations and outline our own experiences with some of these bridging technologies. Section 2 will give an insight to some of the bridging technologies introduced. Section 3 discusses our configuration and set-up of the bridging technologies. Section 4 presents some results of our implementation. Section 5 discusses some PHP Wiki applications tested with the PHP-JavaBridge. While Section 6, provides an insight to our future work involving a Single Sign-On (SSO), which interfaces to existing web applications from a portlet. Finally, Section 7 concludes the paper.

2. Bridging Technologies

In this section we discuss and outline the reference implementation of JSR-223, then we move on to detail both PHP-JavaBridge and the PortletBridge-portlet in terms of architecture, functionality and features.

2.1 JSR-223

The JSR-223 specification is designed to allow scripting language pages to be used in Java server-side applications and vice versa. It basically allows scripting languages to access Java classes. JSR-223 is currently a public draft specification and may therefore change. The material presented in this section is based on the draft specification. In order to assess JSR-223's reference implementation we examine it in relationship with PHP as the scripting language and Apache Tomcat as the servlet container.

JSR-223 introduces a new scripting API `javax.script` that consists of interfaces and classes, which define the scripting engines and a framework for their use in Java. JSR-223 also provides a number of scripting engines for different script based languages. Currently, under the reference implementation version 1.0 [11], there are servlet engines for Groovy, Rhino and PHP. The `GroovyScriptEngine`, `RhinoScriptEngine` and `PHPScritpEngine` classes in the API represent the servlet engines, respectively.

PHP uses a Server API (SAPI) module, which defines interactions between a Web server and PHP. PHP 4 includes Java capabilities provided either by a servlet SAPI extension module or by integrating Java into PHP. The Java SAPI module enables PHP to be run as a servlet while using Java Native Interface (JNI) to communicate with Java. However, unlike PHP 4, PHP 5 does not include the Java integration support in its current release, although it is available under the development branch. The version of PHP 5 included with the reference implementation of JSR-223 does not use the more traditional CGI/FastCGI as its server API. Instead, it makes use of the new PHP Java scripting engine.

JSR-223's reference implementation is based on PHP 5.0.1. The current version of PHP is 5.1.2. Even though there is a mismatch, versions of PHP can be compiled for inclusion into JSR-223. Very few PHP extensions are provided by JSR-223. Any additional PHP extensions desired can be compiled against the version of PHP packaged with JSR-223.

With respect to the installation of the JSR-223, Apache Tomcat is configured by the installation script to process `.php` files. When executing PHP applications with JSR-223, the PHP servlet engine calls the PHP interpreter and passes on the script. Although JSR-223 contains a PHP executable binary, it is not used. Instead, the JSR-223 servlet uses JNI for native library files, namely `php.so` or `php.dll` under Windows. The advantage to this method is that the performance of PHP under JSR-223 is similar to that of a native Apache Web server with `mod_php`.

JSR-223 has examples of session sharing between JSP and PHP, amongst others. Thus JSP and PHP applications may exchange or share information via sessions. Both types of session may be attached to a single PHP

application, through the use of the `use_trans_sid` option, resulting in the Web server's response containing both session IDs.

The PHP servlet engine provides additional objects as predefined variables to aid interoperability. JSR-223 requires that objects, such as an HTTP request, should be available from scripting languages. From a PHP script, these predefined variables reflect the same variables available from a JSP page, namely:

- `$request`
- `$response`
- `$servlet`
- `$context`

Although there can be communication between PHP and Tomcat sessions, there are limitations on what a session can hold. A Tomcat session may only hold scalar values, i.e. strings and numbers. It may not contain database connections, file handlers or PHP objects, but may contain Java objects.

Other limitations of the reference implementation are most noticeable when migrating or using existing PHP applications. PHP applications usually make extensive use of predefined variables, however `$_SERVER['REQUEST_URI']` or `$_SERVER['QUERY_STRING']` are not defined, instead the request variables can be found in `$_SERVER['argv']`. Another limitation is relative paths within PHP pages, which can be worked around in two ways. One of which is to set the `include_path` in the `php.ini` initialisation file. The other way is to modify PHP applications to use absolute paths.

Although JSR-223 is still a reference implementation under review, it does make substantial progress towards the interoperability of Java and script-based languages. This brings forth a new dimension of Java interoperability, which can be useful for developers and end users alike. Despite JSR-223's limitations, it does provide foundations for other implementations.

2.2 The PHP-JavaBridge

The PHP-JavaBridge is as the name implies a PHP to Java bridge. The PHP-JavaBridge uses JSR-223 concepts while adding important aspects, such as enabling communications with already existing PHP server installations. The PHP-JavaBridge is described as an XML-based network protocol designed to communicate with native scripting engines, which have a Java or ECMA 335 [13] virtual machine.

The PHP-JavaBridge XML protocol is used to connect to PHP server instances. The PHP server instance can be a J2EE container, such as Apache Tomcat or a PHP enabled Web server, such as Apache or IIS. PHP version 5.1.2 is included with the PHP-JavaBridge as a CGI/FastCGI component, with native library files, such as `php-cgi-i386-liunx.so` for Linux. The bridge is not limited to the included CGI environment. The operating system's native installation of a PHP CGI/FastCGI can also be used, or an alternative set of binaries can replace those included. Although PHP is the focus of the PHP-JavaBridge project, it does include other features, including the ability to contain PHP scripts with JSF (since version 3.0 of the PHP-JavaBridge) and RMI/IIOP. Examples of these capabilities are included in the standard distribution.

Unlike the reference implementation of JSR-223, the PHP-JavaBridge does not use JNI. Instead, HTTP instances are allocated by the Web server or from the Java/J2EE backend. The PHP instances communicate using a "continuation passing style", which means that if PHP instances were to crash it would not takedown the Java server or servlet at the backend. Because the PHP-JavaBridge communicates via a protocol, it does have a certain amount of additional functionality. There can be more than one HTTP server routed to a single PHP-JavaBridge or each HTTP server can have its own PHP-JavaBridge. When the PHP-JavaBridge is running in a J2EE environment, it is also possible to share sessions between JSP and PHP. Although the PHP-JavaBridge can use Novell MONO or Microsoft .NET as a backend, this is beyond the focus of this section.

As mentioned, the PHP-JavaBridge is flexible with regards to the type of PHP server or communication method that it uses. The PHP-JavaBridge also has four operating modes, some of which implement practical solutions for production servers. The different operating modes of the bridge have been categorised as:

- Request.
- Dynamic.
- System.
- J2EE.

When the PHP-JavaBridge is operated in Request mode, the bridge is created and

destroyed on every request or response. While the bridge is running in Dynamic mode the bridge starts and stops synchronously with a pre-defined HTTP server. System mode is when the bridge is run as a system service. This mode is installed as an RPM under Red Hat Enterprise Linux. Finally, the J2EE mode is when the bridge is installed in a J2EE server, such as Apache Tomcat. Other modes and configurations are possible, though not discussed, are here [5], including executing a standalone bridge and configurations for Security Enhanced Linux distributions.

When using a J2EE environment, PHP applications can be packaged with a PHP-JavaBridge into a `.war` archive. The archive will contain the configuration of the bridge, the PHP scripts and if needed a CGI environment. This makes Web applications for Java, based on PHP, easy to deploy.

The PHP-JavaBridge does not encounter the same issues with PHP extensions, as the JSR-223. This is due to the configuration flexibility of the PHP environment. When utilising an existing PHP installation, any extension module can be installed in the normal manner, with only little or no configuration required of the PHP-JavaBridge. However, the ease of integrating an existing PHP application into a J2EE environment can be hampered by the application itself. A PHP application, in some cases, is not fully aware of the J2EE environment, such as host names containing the additional port in the URL. Some predefined PHP variables are also not available in certain circumstances, or are not correct under the PHP-JavaBridge environment.

The performance of the PHP-JavaBridge [5] shows that the overall execution time is similar to that of native scripting engines executed from the command line with `jrunscript`. The results also show that there is a performance overhead when using the PHP-JavaBridge communication protocol. However, the overhead introduced is not substantial and considering that communications between different hosts is only 50% slower than communications on a single host, it is fair to ignore the overheads in favour of flexibility.

Developers using either JSR-223 or the PHP-JavaBridge can produce hybrid Java and PHP applications. Session sharing between JSP and PHP is also available with both technologies. However, the PHP-JavaBridge offers greater flexibility and integration when communicating with existing PHP applications or servers.

2.3 The PortletBridge-Portlet

While JSR-223 and the PHP-JavaBridge concentrate on the interoperability of scripting languages and Java. The PortletBridge-portlet focuses on creating a generic method of allowing Web sites to be viewed via a JSR-168 compliant portlet. Typically, portals, such as uPortal provide the means to view Web sites via Iframes. Yet, the JSR-168 specification does not recommend their use, as there is no control over the Iframe for the portal. Typically, links within an Iframe are external from the portal environment, which can result in inconsistencies with a user's experience.

The PortletBridge-portlet is also known as a 'Web Clipping Portlet'. The PortletBridge-portlet proxies Web site content and renders the downstream content into a portlet. An interesting concept from the PortletBridge-portlet is using XSLT to control or transform the rendering of the downstream content. This is ideal for portlet environments, as portlets do not typically cover an entire page. Developers can select the parts they wish to render in a portlet, which is unlike an Iframe. These differences are also highlighted with the ability to resize the PortletBridge-portlet, which is also not possible in an Iframe.

The PortletBridge-portlet also allows developers to control, which links within the downstream content are proxied and which are not. Images or Flash animations from remote resources can also be proxied by the PortletBridge-portlet. A regular expression is used to define the links to proxy within the portlet and which links are external to the portlet. The rewriting of the downstream content is not limited to links. JavaScript and CSS can also be handled by the XSLT or with regular expressions. By default, only the body of a Web page is rendered by the PortletBridge into a portlet. This follows the JSR-168 specification in which no header or footer tags are allowed into portlets. However, it is still possible to manipulate the content of a page's header/footer with XSLT and have it rendered into the portlet.

The PortletBridge-portlet uses the CyberNecko HTML parser [14], which allows HTML pages to be accessed via XML interfaces. It is built around the Xerces Native Interface (XNI) that has the ability to communicate "streaming" documents. The PortletBridge-portlet also

defines a "memento" that holds user states, such as cookies. The memento is shared between the PortletBridge portlet and the PortletBridge servlet. For this interaction to occur the J2EE container, in this case Apache Tomcat needs to have `crossContext` enabled.

Currently the PortletBridge-portlet lacks the features to support complex XHTML and is a relatively young open source project. Nevertheless, it does present a viable option for making Web sites or applications available through a portlet. In addition, due to the adherence of the PortletBridge-portlet to the JSR-168 specification, it is also possible to use this technology in various different portals or even consume the portlet with Web Services for Remote Portlets (WSRP) [15]. Because the PortletBridge-portlet is a generic solution that is confined to portals, it is also a practical option to combine the PortletBridge with other bridging technologies such as the PHP-JavaBridge to provide a fully featured application environment.

3. BibAdmin Portlet Configuration

Non-Java applications, in our case, PHP, need to be redesigned or re-implemented with Java for use as a portlet. Using the techniques described in this section, a developer can reduce the amount of work needed to have a PHP application available as a portlet under a J2EE environment.

Our configuration consisted of the following components:

- Debian Linux with PHP5-CGI binaries,
- A PHP application,
- MySQL server 4.1.x,
- Apache Tomcat 5.5.16,
- PHP-JavaBridge 3.0.8r3,
- PortletBridge-portlet CVS,
- GridSphere 2.1.x.

Our installation of PHP included the MySQL extension, as it is common for PHP applications to be part of a LAMP/WAMP stack. An arbitrary multi-user application was chosen, namely BibAdmin 0.5 [16]. BibAdmin is a Bibtex bibliography server. The need for a multi-user application was to match the security framework of the portal and is required for future work outlined in Section 5. GridSphere is a JSR-168 compliant portal with good user support, which we have used extensively in other projects.

The PHP-JavaBridge was installed in Apache Tomcat as a `.war` file. The `php-servlet.jar` and `JavaBridge.jar` were also placed in `$CATALINA_HOME/shared/lib` as per the installation instructions [5]. The PHP-CGI binary and the MySQL PHP extension were used from the Debian Linux installation, instead of those originally included with the PHP-JavaBridge. This then provided a feature rich PHP environment to applications running with the PHP-JavaBridge.

When using PHP applications under the PHP-JavaBridge the PHP's configuration file (`php.ini`) is not in a default location. Instead it is located under `$CATALINA_HOME/webapps/JavaBridge/WEB-INF/cgi`, referenced as `php-cgi-i386-linux.ini`. Any PHP configurations are placed in the PHP configuration file, such as the configuration of the MySQL extension.

PHP 5 has some differences compared to PHP 4, which can lead to application incompatibilities. BibAdmin is a PHP 4 application, which was migrated to PHP 5 [17]. Certain PHP functions used in BibAdmin required modifications to reflect API changes in PHP 5, such as `mysql_fetch_object()`. Other changes included some PHP predefined variables. The PHP variable `$_SERVER['SERVER_NAME']` was changed to `$_SERVER['HTTP_HOST']`. This change was used to reflect the additional port number in the URLs needed for BibAdmin administrative tasks when creating a new user. Another useful PHP variable was `$_SERVER['PHP_SELF']`, which not only provides the PHP page, but also the Web application directory, i.e. `/BibAdmin-0.5/index.php`. BibAdmin under a normal PHP 5 setup was compared to BibAdmin under the PHP-JavaBridge. No differences were found in terms of functionality or usage.

The PortletBridge-portlet is typically installed from a Web archive. However, with GridSphere, the PortletBridge-portlet was first expanded into a directory. This was due to a missing root directory in the PortletBridge-portlet package. The created directory is copied into Apache Tomcat's `webapps` directory. Some additional changes to the `web.xml` file were needed for the PortletBridge-portlet to operate correctly in GridSphere. In addition, the GridSphere UI tag library is required.

A simple XSL style sheet was used that defines what transformations are carried out on the

downstream content from a Web site. In the Edit mode of the portlet, the configuration of the portlet can be altered. The preferences include proxy server authentication, the initial URL to load and XSL style sheets. The Edit mode of the PortletBridge-portlet is only used for development purposes. The end user would not generally use these preferences. The portlets proxy configurations are useful for larger or more complex networks. Different proxy authentication methods can be used, such as Windows NT LAN Manager (NTLM).

BibAdmin was configured with a MySQL database connection setting and packaged as a `.war` file. Included in the Web archive's `WEB-INF` was the CGI components, `web.xml` and any PHP-JavaBridge dependencies.

Figure 1, shows the final layout of the configuration for reusing a Web application in a portlet under a J2EE environment.

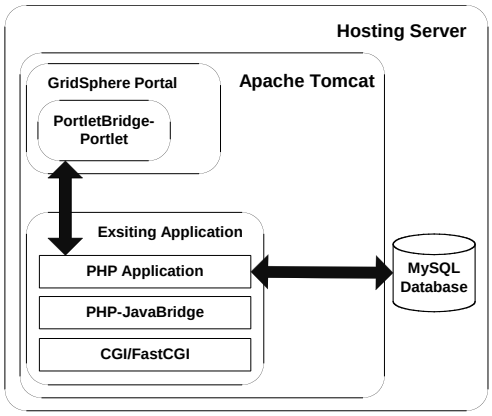


Figure 1: BibAdmin Portlet Configuration

The flexibility offered by the PHP-JavaBridge ensures that it is also possible to use the more common Apache Web server and PHP combination. The Apache Web server could also be located on the same or a different host. Communications with the Apache Web server would be with the XML protocol via sockets. This configuration can be de-coupled even further by having the portal and PortletBridge-portlet located on a different host. It was decided that a J2EE environment would provide the most challenges and thus best demonstrate the set-up.

4. The BibAdmin Portlet Results

It was observed that the PHP application, BibAdmin, was equally functional under the PHP-JavaBridge as it was under a normal Apache Web server. Although, this is only true after the minor changes needed to BibAdmin to make it aware of its different environment. Figure 2, shows the BibAdmin portlet login page. Figure 3, is another screenshot of the BibAdmin performing as a portlet.

Some parts of the PortletBridge-portlet are not complete, such as XHTML support. This hampers the number of sites that can be proxied by the PortletBridge. Currently PHP logins are also an issue. The PortletBridge developers have been contacted about this problem and it is under investigation. However, the login issue is not completely relevant to our needs, as it does not fit in with the portals security framework. A user should be automatically be logged into BibAdmin with the appropriate credentials, provided by the portal security framework. This area of research is discussed further in our future work under Section 6.

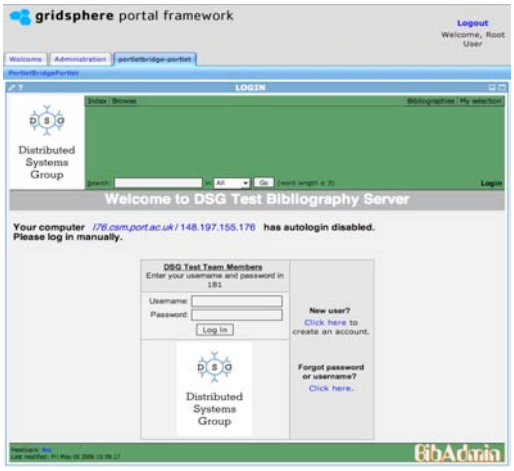


Figure 2: The BibAdmin Portlet

5. PHP Wiki Systems

This technique of reusing Web applications as portlets has been applied to other PHP applications. A Wiki offers an online resource, which allows multiple users to add and edit content collectively. It is also a suitable application to include in a portal and to test our bridging system. PHP Wiki applications include, PHPWiki [18] and DokuWiki [19].

Figure 3: The BibAdmin Portlet - BibTex Use

Both of these Wikis work under the PHP-JavaBridge, however they do not render correctly within the PortletBridge-portlet. This rendering error is due to the lack of support in the PortletBridge-portlet for XHTML. Figure 4 and Figure 5 show screenshots of PHPWiki and DokuWiki running under the PHP-JavaBridge, respectively.

Figure 4: PHPWiki with the PHP-JavaBridge

Figure 5: DokuWiki with the PHP-JavaBridge

As can be seen from figures that PHP applications work under the J2EE environment. The errors seen in the PortletBridge-portlet are primarily focused on the rendering. Not that it does not render, just that it is not usable. Once the PortletBridge-portlet has included XHTML support, these issues should be resolved.

6. Future Work

Our future work will concentrate on integrating PHP application further within portal frameworks. One area that we feel is important to develop urgently is a generic Single-Sign-On (SSO) system. Users can only access Web application portlets after being authenticated, by the portals security framework. It should then not be necessary to re-authenticate the user again with the Web application. Instead, the SSO system will use the security framework in the portal to provide an automated and trusted login for the Web application. Our SSO system will use a combination of the PHP-JavaBridge and the PortletBridge-portlet to pass information between a Portal and the PHP application. With this system a user's experience of Web applications via portlets will be transparent.

7. Conclusions

This paper has presented Java and portlet bridging technologies. Both types of bridging technologies have had their features and capabilities reviewed. We have also demonstrated how to reuse existing Web applications as portlets. Using a combination of these technologies, it is also possible to merge and manipulate the best features of Java, and scripting based languages. Developers can also concentrate on an applications usability rather than integration of technologies. This technique provides rapid portlet development resulting in highly usable portlets.

Portal developers can use the techniques presented, to not only provide new and existing tools along side within a common environment, but also allow applications to be used as a portlet and Web application simultaneously.

References

[1] JSR -168 specification, http://www.jcp.org/en/jsr/detail?id=168
[2] PHP, http://www.php.net
[3] Ruby on Rails, http://www.rubyonrails.org
[4] JSR-223: Scripting for the Java Platform, http://www.jcp.org/en/jsr/detail?id=223
[5] PHP-JavaBridge, http://php-java-bridge.sourceforge.net/
[6] Apache Portals, Portals Bridges, http://portals.apache.org/bridges/
[7] Apache Struts, http://apache.struts.org
[8] Struts Bridge, http://portals.apache.org/bridges/multiproject/portals-bridges-struts/index.html
[9] GridSphere Portal, http://www.gridsphere.org/
[10] PortletBridge-Portlet, http://www.portletbridge.org/
[11] JSR-223 Reference Implementation, http://jcp.org/aboutJava/communityprocess/pr/jsr223/index.html
[12] Rhino, http://www.mozilla.org/rhino/

[13] ECMA-335 standard, http://www.ecma-international.org/publications/standards/Ecma-335.htm
[14] CyberNeko HTML parser, http://people.apache.org/~andyc/neko/doc/html/index.html
[15] WSRP, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp
[16] BibAdmin, http://www-sop.inria.fr/axis/personnel/Sergiu.Chelcea/software.php?soft=bibtex
[17] PHP Manual, Migrating to PHP 5, http://php.ftp.cvut.cz/manual/en/migration5.php
[18] PHPWiki, http://phpwiki.sourceforge.net/phpwiki/
[19] DokuWiki, http://wiki.splitbrain.org/wiki:dokuwiki

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

GANGA: A Grid User Interface for Distributed Data Analysis

K. Harrison
Cavendish Laboratory, University of Cambridge, CB3 0HE, UK

C.L. Tan
School of Physics and Astronomy, University of Birmingham, B15 2TT, UK

D. Liko, A. Maier, J.T. Moscicki
CERN, CH-1211 Geneva 23, Switzerland

U. Egede
Department of Physics, Imperial College London, SW7 2AZ, UK

R.W.L. Jones
Department of Physics, University of Lancaster, LA1 4YB, UK

A. Soroko
Department of Physics, University of Oxford, OX1 3RH, UK

G.N. Patrick
Rutherford Appleton Laboratory, Chilton, Didcot, OX11 0QX, UK

Abstract

Details are presented of GANGA, the Grid user interface being developed to enable large-scale distributed data analysis within High Energy Physics. In contrast to the standard LCG Grid user interface it makes transparent most of the Grid technicalities. GANGA can also be used as a frontend for smaller batch systems thus providing a homogeneous environment for the data analysis on inhomogeneous resources. As a clear software framework GANGA offers easy possibilities for extension and customization. We report on current GANGA deployment, and show that by creating additional pluggable modules its functionality can be extended to fit the needs of other grid user communities.

1. Introduction.

GANGA [1] is an easy-to-use frontend for job definition and management, implemented in Python [2]. It is being developed to meet the needs of a Grid user interface within the ATLAS [3] and LHCb [4] experiments in High Energy Physics, and is a key piece of their distributed-analysis systems [5,6].

ATLAS and LHCb will investigate various aspects of particle production and decay in high-energy proton-proton interactions at the Large Hadron Collider (LHC) [7], due to start operation at the European Laboratory for Particle Physics (CERN), Geneva, in 2007. Both experiments will require processing of data volumes of the order of petabytes per year, and will rely on computing resources distributed across multiple locations. The experiments' data-processing applications, including simulation, reconstruction and physics analysis, are based on the GAUDI/ATHENA C++ framework [8]. This provides core services, such as message logging, data access, histogramming; and allows run-time configuration.

GANGA deals with configuring the ATLAS and LHCb applications, allows switching between testing on a local batch system and large-scale processing on the Grid, and helps keep track of results. All this information is contained within an object called GANGA job.

Jobs for simulation and reconstruction typically use GAUDI/ATHENA software that results from a coordinated, experiment-wide effort, and is installed at many sites. The person submitting the jobs, possibly a production manager, performs the job configuration, which involves selecting the algorithms to be run, defining the algorithm properties and specifying inputs and outputs. The situation is similar for an analysis job, except that the physicists running a given analysis will usually want to load one or more

algorithms that they have written themselves, and so use code that may be available only in an individual physicist's work area. Another major difference between analysis and production jobs consists in the amount of input data they process. As a rule an analysis job requires gigabytes or even terabytes of data collected in so called datasets and distributed among many storage elements around the globe. Discovery of dataset locations is done trough recourse to various metadata and file catalogues and hence a mechanism for data discovery has to be included as an integral part of performing an analysis.

In Section 2 we give detailed description of a GANGA job. Section 3 describes the overall architecture of GANGA including the persistency and plugin systems. GANGA functionality and different aspects of user interface are summarized in Section 4. In Section 5 we report on current GANGA use within ATLAS and LHCb, and finally in Section 6 we provide more details about GANGA as a software framework and show ways how it can be extended to satisfy needs of other potential users.

2. Job Representation

A job in Ganga is constructed from a set of building blocks (Fig. 1). All jobs must specify the software to be run (application) and the processing system (backend) to be used. Many jobs will specify an input dataset to be read and/or an output dataset to be produced. Optionally, a job may also define functions (splitters and mergers) for dividing a job into subjobs that can be processed in parallel, and for combining the resultant outputs. In this case after splitting the job becomes a master job and provides a single point of access for all its subjobs.

Different types of application, backend, dataset, splitter and merger are implemented as plugin classes (see Section 6). Each of these has its own schema, which describes the configurable properties and their meanings. Properties are defined both to permit the user to set values defining the operations to be performed within a job, and to store information returned by the processing system, allowing tracking of job progress.

Applications for which plugins have been written include a generic Executable application, the ATLAS ATHENA application, and the GAUDI-based applications of LHCb.

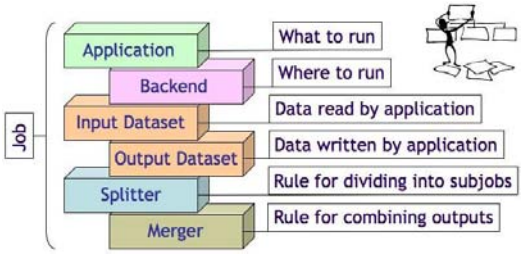


Figure 1: Building blocks for constructing a GANGA job.

The backend plugins cover generic distributed systems, such as the LHC Computing Grid (LCG) [9] and gLITE [10]; experiment-specific distributed systems, such as DIRAC [11] in LHCb; and local batch systems, including LSF, PBS and Condor.

3. Architecture

The functionality of GANGA is divided between components (Fig.2). GANGA Core links them together and performs most common tasks. It is represented by Application and Job Managers, Job Repository, and File Workspace. All components communicate via the GANGA Public Interface (GPI), which is designed to help GANGA users and external developers who are not proficient in Python to write their scripts and plugin modules more easy. The Client allows access to GPI commands in any of three ways: through a shell -- the Command Line Interface in Python (CLIP); using GPI scripts, or through a Graphical User Interface (GUI).

3.1 Application Manager

The Application Manager deals with defining the task to be performed within a job, including the application to be run, the user code to be executed, the values to be assigned to any configurable parameters, and the data to be processed. GANGA calls the Application Manager for every job before its submission. The duty of the Application Manager is to return a configuration object which contains backend-independent information. In case of split jobs the operation of the Application Manager is optimised, so that it performs only those configuration actions for subjobs, which are not factorized in the master job. For jobs with a complicated configuration this can speed up the time required for job submission by orders of magnitude.

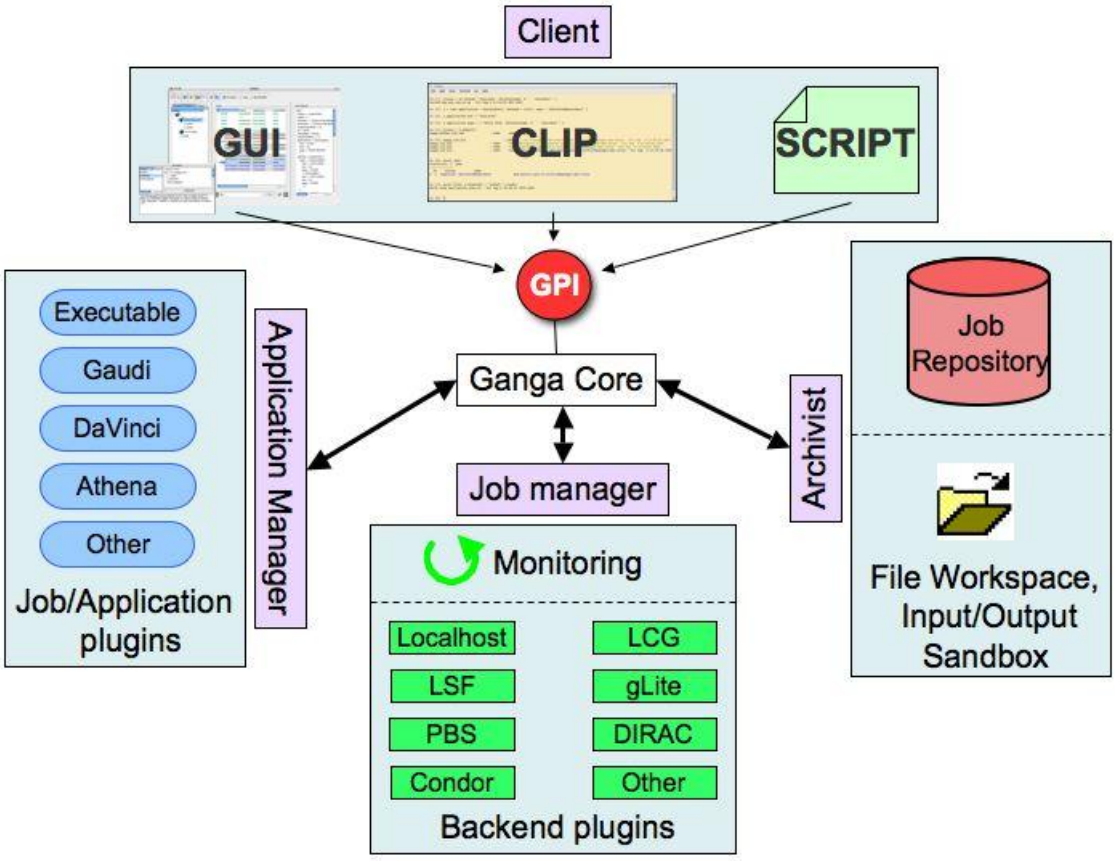


Figure 2: Schematic representation of the GANGA architecture. The main functionality is divided between Application Manager, Job Manager and Archivist, and is accessed by the Client through the GANGA Public Interface (GPI). The client can run the Graphical User Interface (GUI), the Command-Line Interface In Python (CLIP) or GPI scripts.

3.2 Job Manager

The Job Manager takes care of any job splitting requested, packages up required user code, performs submission to the backend, monitors job progress, and retrieves output files when jobs complete. It takes as input the configuration object prepared by the Application Manager and calls the application runtime handler – a class which is responsible for the backend-specific part of the application configuration. Such a configuration mechanism reduces the number of required software modules because application runtime handlers are normally compatible with more than one type of application. It also allows dynamic connection to a new backend, and running the same application on different Grid or batch systems.

The Job Manager acquires information about job status with the help of the Monitoring component that in turn polls plugin modules representing backends to which jobs were submitted.

3.3 Job Repository

Acting as a simple bookkeeping system the Job Repository stores all GANGA jobs. The Job Repository and File Workspace provide persistency for jobs and their input/output sandboxes so that GANGA can be restarted having access to the same information. If necessary this information can be searched for particular jobs according to job metadata which is also stored in the Job Repository.

The Job Repository can be either local or remote. Both types of repository have the same interface, so there is no difference from the programmatic point of view. However they differ substantially in implementation and in their usage.

The local Job Repository is a lightweight database written entirely in Python so that it can be easy run on any computer platform. It is mainly designated for individual users and provides possibility to work with GANGA off-line, for example prepare jobs for submission until on-line resources become available. In contrast, the remote Job Repository is a service centrally maintained for the need of many users. It gives the advantage of job access from any location where internet connection is present. Current implementation of the remote Job Repository is based on the AMGA metadata interface [12]. It supports secure connection and user authentication and authorization based on Grid proxy certificates.
The performance tests of both the local and remote repositories show good scalability for up to 10 thousand jobs per user, with the average time of job creation being 0.2 and 0.4 second for the local and remote repository correspondingly.

3.4 Plugin modules

GANGA comes with a variety of plugin modules representing different types of applications and backends. These modules are controlled correspondingly by the Application and Job Manager. Such a modular design allows the functionality to be extended in an easy way to suit particular needs as described in Section 6.

4. User view

User interacts with GANGA trough the Client and configuration files.

4.1 Configuration

GANGA has default parameters and behaviour that can be redefined at startup using one or more configuration files, which use the syntax understood by the standard Python [2] ConfigParser module. Configuration files can be introduced at the level of system, group or user, with each successive file processed able to override settings from preceding files. The configuration files allow selection of the Python packages that should be initialised when GANGA starts, and consequently of the modules, classes, objects and functions that are made available through the GPI. They also allow modification of the default values to be used when creating objects from plugin classes, and permit actions such as choosing the log level for messaging, specifying the location of the job repository, and changing certain visual aspects of GANGA.

4.2 Command Line Interface in Python

GANGA's Command Line Interface in Python (CLIP) provides for interactive job definition and submission from an enhanced Python shell, IPython [13], with many nice features. A user needs to enter only a few commands to set application properties and submit a job to run the application on a chosen backend, and switching from one backend to another is trivial. CLIP includes possibilities for organising jobs in logical files, for creating job templates, and for exporting jobs to GPI scripts. Exported jobs can be freely edited, shared with others, and/or loaded back into GANGA. The CLIP is especially useful for learning how GANGA works, for one-off job-submissions, and -- particularly for developers -- for understanding problems if anything goes wrong. A realistic scenario of full submission of an analysis job in LHCb is illustrated in Fig 3.

4.3 GPI scripts

GPI scripts allow sequences of commands to be executed in the GANGA environment, and are ideal for automating repetitive tasks. GANGA includes commands that can be used outside of the Python/IPython environment to create GPI scripts containing job definitions; to perform job submission based on these scripts, or on scripts exported from CLIP; to query job progress; and to kill jobs. Working with these commands is similar to working with the commands typically encountered when running jobs on a local batch system, and for users can have the appeal of being immediately familiar.

4.4 Graphical User Interface

The GUI (Fig. 4) aims to further simplify user interaction with GANGA. It is based on the PyQt [14] graphics toolkit and GPI, and consists of a set of dockable windows and a central job monitoring panel. The main window includes:

- a toolbar, which simplifies access to all the GUI functionality;
- a logical folders organiser implemented in a form of job tree;
- a job monitoring table, which shows the status of user jobs selected from the job tree;
- a job-details panel, which allows inspection of job definitions.

The Job Builder window has a set of standard tool buttons for job creation and modification. It also has a button for dynamic export of plugin methods, job attribute value entry widgets, and a job attribute tree view. The scriptor window allows the user to execute arbitrary Python

```
# Define an application object
dv = DaVinci(version = 'v12r15',
      cmt_user_path = '~/public/cmt',
      optsfile = 'myopts.opts')

# Define a dataset
dataLFN = LHCbDataset(files=[
'LFN:/lhcb/production/DC04/v2/00980000/DST/Presel_00980000_00001212.dst' ,
:
'LFN:/lhcb/production/DC04/v2/00980000/DST/Presel_00980000_00001248.dst'])

# Put application, backend, dataset and splitting strategy together in a job
# Dirac is the name of the LHCb submission system to the Grid.
j = Job(application=dv,
       backend=Dirac(),
       inputdata=dataLFN,
       splitter=SplitByFiles())

# Submit your complete analysis to the Grid.
j.submit()
```

Figure 3: A illustration of the set of CLIP commands for submission of a job to perform analysis in LHCb. While definition of the application includes elements specific to the LHCb experiment, the definition of the job follows the same structure everywhere.

scripts and GPI commands, maximising flexibility when working inside the GUI. Message from GANGA are directed to a log panel.
By default, all windows are shown together in a single frame, but because they are dockable they can be resized and placed according to the tastes of the individual user. Job definition and submission is accomplished through mouse clicks and form completion, with overall functionality similar to CLIP

5. Deployment

Although still in the development phase, GANGA already has functionality that makes it useful for physics studies. Tutorials for ATLAS and LHCb have been held in the UK and at CERN, and have led to GANGA being tried out by close to a 100 people.
Within LHCb Ganga has seen regular use for distributed analysis since the end of 2005. There are about 30 users at present of the system and performance has been quite satisfactory. In a test an analysis which took about 1s per event ran over 5 million events. For the analysis on the Grid the job was split into 500 subjobs. It was observed that 95% of the results had returned within less than 4 hours while the remaining 5% initially failed due to a problem at a CE. These jobs were automatically restarted and the full result was available within 10 hours. The fraction of the analysis completed as a function of time is illustrated in Fig. 5. Recently more than 30 million events have been processed with the success rate exceeding 94%. Further details on the specific use of Ganga within LHCb can be found in [15].
In the last months GANGA found a new application outside the physics community. In particular, it was used for job submission to the EGEE and associated computing grids within the activities towards the avian flu drug discovery [16], and for obtaining optimum planning of the available frequency spectrum for the International Telecommunication Union [17].

6. Extensions

Despite similarities between ATLAS and LHCb they have some specific requirements and use cases for the Grid interface. In order to deal with these complications GANGA was designed as a software framework having a pluggable component structure from the very beginning. Due to this architecture GANGA readily allows extension to support other application types and

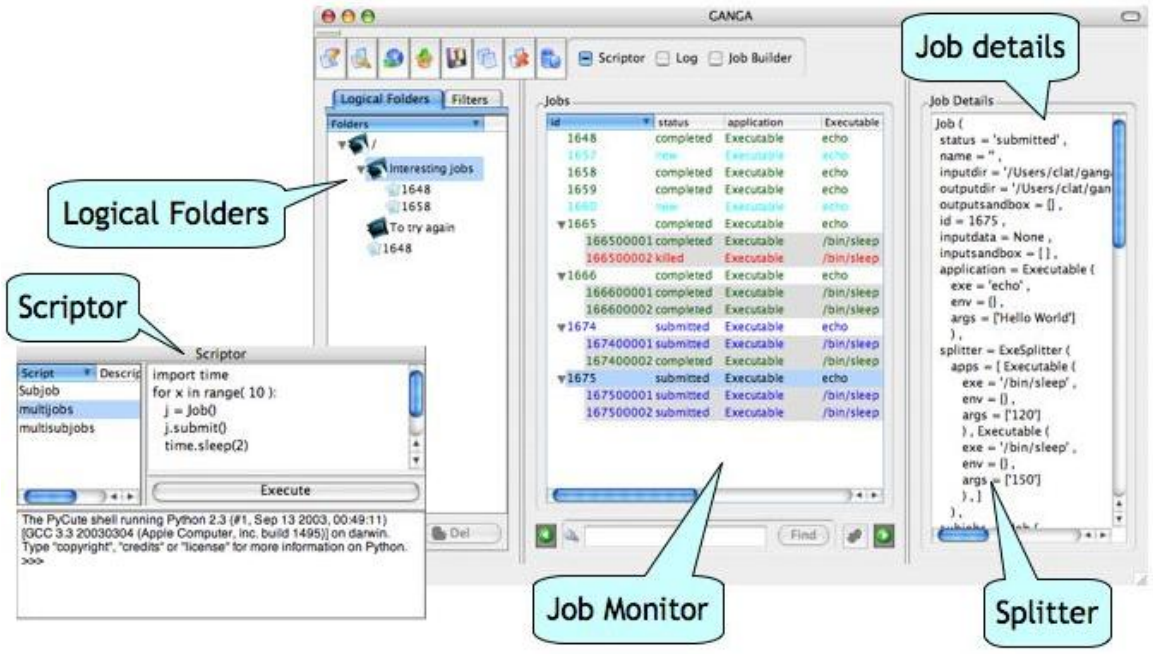


Figure 4: Screenshot of the GANGA GUI, showing: a main window (right), displaying job tree, monitoring panel and job details; and an undocked scriptor window (left).

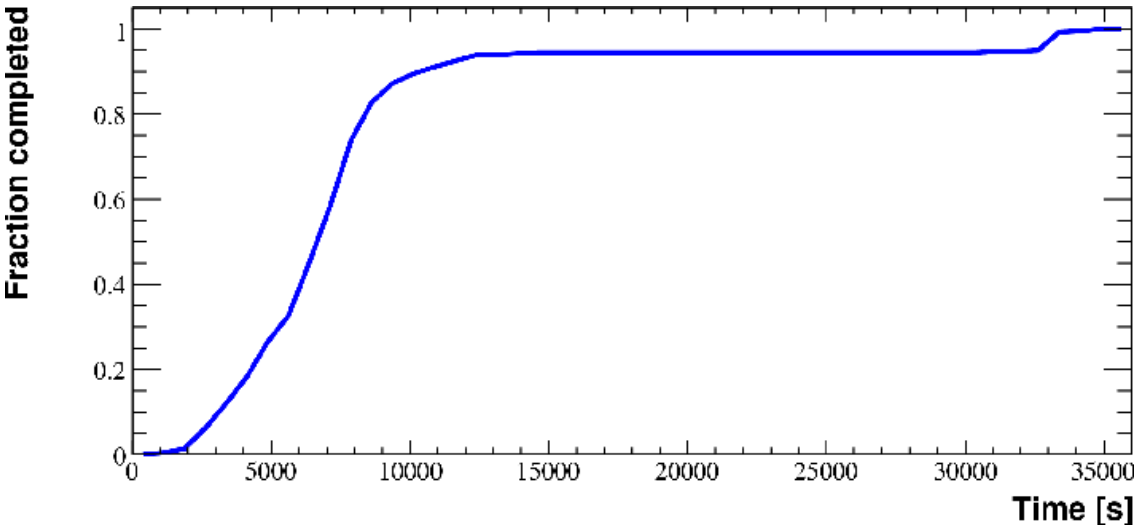


Figure 5: The fraction of the data analysis completed as a function of time

backends that can be found outside these two experiments. In particular, GANGA could be very useful for any computing task to be executed in a Grid environment that depends on sequential analysis of large datasets.

In practice plugins associated with a given category of job building block inherit from a common interface class - one of IApplication, IBackend, IDataset, ISplitter and IMerger as illustrated in Fig. 6. The interface class documents the required methods -- a backend plugin, for example must have submit and kill methods -- and contains default (often empty) implementations.

The interface classes have a common base class, GangaObject, which provides a persistency mechanism and allows user default values for plugin properties to be set via a configuration

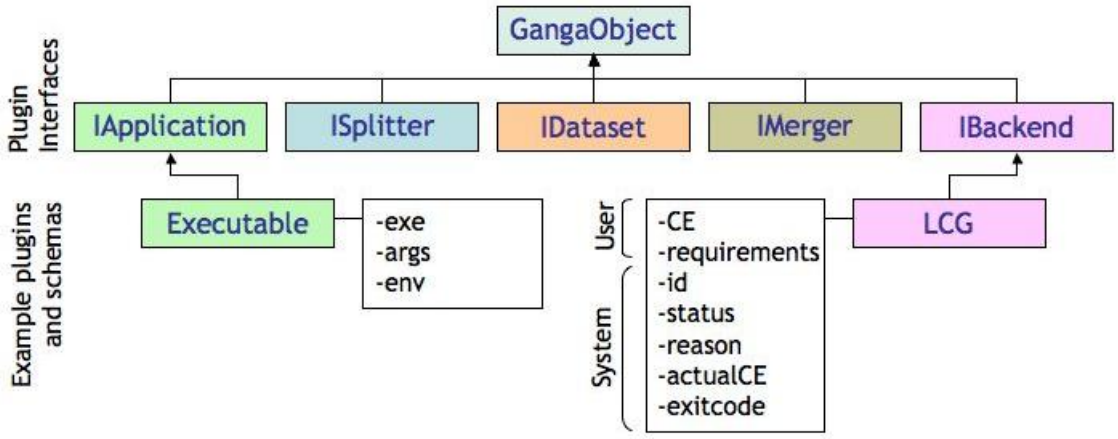


Figure 6: GANGA plugins. Plugins for different types of application, backend, dataset, splitter and merger inherit from interface classes, which have a common base class. Schemas for the Executable application and for the LCG backend are shown as examples.

file. Another function of the GangaObject class is to introduce a uniform description of all methods and data members visible to the user. Such a description called GANGA Schema allows GUI and CLIP representations of the plugin classes to be build automatically. Thus a GANGA developer can add a new plugin without special knowledge of the GUI and CLIP frameworks.

7. Conclusions

GANGA is being developed as a Grid user interface for distributed data analysis within the ATLAS and LHCb experiments. In contrast to many existing web portals [18] providing access to the Grid GANGA is a set of user tools running locally and therefore can offer more enhanced functionality. For example, it simplifies configuration of applications based on the GAUDI/ATHENA framework used in ATLAS and LHCb; allows trivial switching between testing on a local batch system and running full-scale analyses on the Grid, hiding Grid technicalities; provides for job splitting and merging; and includes automated job monitoring and output retrieval. GANGA offers possibilities for working in an enhanced Python shell, with scripts, and through a graphical interface.

Although specifically addressing the needs of ATLAS and LHCb for running applications performing large-scale data processing on today's Grid systems, GANGA has a component architecture that can be easily adjusted for future Grid evolutions. As compared with similar Grid user interfaces [19] developed within other High Energy Physics experiments GANGA has a clear application programming interface that can be exploited as an "engine" for implementing application-specific portals in different science areas. Written in Python GANGA can also benefit from more simple integration of new scientific applications as compared with the Java based Grid user interfaces such as Espresso [20].

GANGA has been tried out by close to 100 people from ATLAS and LHCb, and growing number of physicists use it routinely for running analyses on the Grid, with considerable success. Expansion of GANGA outside the High Energy Physics community was reported recently.

Acknowledgements

We are pleased to acknowledge support for the work on GANGA from GridPP in the UK and from the ARDA group at CERN. GridPP is funded by the UK Particle Physics and Astronomy Research Council (PPARC). ARDA is part of the EGEE project, funded by the European Union under contract number INFSO-RI-508833.

References

[1] http://ganga.web.cern.ch/ganga/
[2] G.van Rossum and F.L. Drake, Jr. (eds.), Python Reference Manual, Release~2.4.3 (Python Software Foundation, 2006); http://www.python.org/
[3] ATLAS Collaboration, Atlas - Technical Proposal, CERN/LHCC94-43 (1994); http://atlas.web.cern.ch/Atlas/

[4] LHCb Collaboration, LHCb - Technical Proposal, CERN/LHCC98-4 (1998); http://lhcb.web.cern.ch/lhcb/
[5] D. Liko *et al*., The ATLAS strategy for Distributed Analysis in several Grid infrastructures, in: Proc. 2006 Conference for Computing in High Energy and Nuclear Physics, (Mumbai, India, 2006)
[6] U. Egede *et al*., Experience with distributed analysis in LHCb, in: Proc. 2006 Conference for Computing in High Energy and Nuclear Physics, (Mumbai, India, 2006)
[7] LHC Study Group, The LHC conceptual design report, CERN/AC/95-05 (1995); http://lhc-new-homepage.web.cern.ch/lhc-new-homepage/
[8] P. Mato, GAUDI - Architecture design document, LCHb-98-064 (1998); http://proj-gaudi.web.cern.ch/proj-gaudi/welcome.html; http://atlas-computing.web.cern.ch/atlas-computing/packages/athenaCore.php
[9] http://lcg.web.cern.ch/lcg/
[10] http://glite.web.cern.ch/glite/
[11] A. Tsaregorodtsev *et al.*, DIRAC -- the LHCb Data Production and Distributed Analysis system, in: Proc. 2006 Conference for Computing in High Energy and Nuclear Physics, (Mumbai, India, 2006)
[12] http://project-arda-dev.web.cern.ch/project-arda-dev/metadata/
[13] http://ipython.scipy.org
[14] http://www.riverbankcomputing.co.uk/pyqt
[15] http://ganga.web.cern.ch/ganga/user/html/LHCb/
[16] http://www.eu-egee.org/press_releases
[17] http://indico.cern.ch/contributionDisplay.py?contribId=34&sessionId=11&confId=286
[18] http://panda.physics.gla.ac.uk/; https://genius.ct.infn.it/; https://grid.ucla.edu:9443/gridsphere/gridsphere
[19] http://cmsdoc.cern.ch/cms/ccs/wm/www/Crab/; http://alien.cern.ch/twiki/bin/view/AliEn/Home; http://projects.fnal.gov/samgrid/
[20] http://freshmeat.net/projects/gridespresso

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Grid Enabled Data Fusion for Calculating Poverty Measures.

Simon Peters[1], Pascal Ekin[2], Anja LeBlanc[2], Ken Clark[1] and Stephen Pickles[2]

[1]School of Social Sciences & [2]Manchester Computing, University of Manchester

2006

Abstract

This article presents and discusses the motivation, methodology and implementation of an e-Social Science pilot demonstrator project entitled: Grid Enabled Micro-econometric Data Analysis (GEMEDA). This used the National Grid Service (NGS) to investigate a policy relevant Social Science issue: the welfare of ethnic minority groups in the United Kingdom. The underlying problem is that of a statistical analysis that uses quantitative data from more than one source. The application of grid technology to this problem allows one to integrate elements of the required empirical modelling process: data extraction, data transfer, statistical computation and results presentation, in a manner that is transparent to a casual user.

1. Introduction

Economic investigation of the experiences and prospects of Britain's ethnic groups paints a picture of multiple deprivation and disadvantage in areas such as earnings and employment. One consequence of this is that the welfare of such groups is of major policy concern. However, in order to evaluate minority welfare, a requirement for successful policy intervention, one needs to be able to produce appropriate statistical quantities, such as poverty measures.

The full modelling process required for this goes beyond an analysis using a single data set, and has components that suggest an e-Research approach is appropriate. The motivation for two of these components, data and statistical modelling, is discussed in section 2, with details of the methodology reported in section 3. Section 3 also deals with Grid adoption issues, and the perspective taken is from the economics discipline, where dedicated e-Science resources, such as equipment and specialised staff, are not readily available. This underlines one of the objectives of the demonstrator. We not only want to show that modern micro-econometric research can be implemented on the Grid, but that it can be done in a manner that builds on existing infrastructure investments (such as the National Grid Service), and develops them.

Another component of our modelling process, results visualization, is briefly discussed in section 4 along with other details of the Grid implementation. Section 5 presents a summary of the substantive analysis, and section 6 concludes.

2. Motivation

Social science researchers in the UK have access to a wide range of survey data sets. These are collected by numerous different agencies, including the government, with different purposes in mind and exhibit considerable variation along a number of dimensions. Only on rare occasions are the needs of social scientists uppermost in the minds of survey designers — hence the topics covered, the sample frames and sizes, the questionnaire formats, data collection methodologies and specific questions asked are extremely diverse. In order to obtain a more complete answer to their research questions, researchers frequently have to use more than one data set. The type of analysis possible, however, is often constrained by the fact that each data set possesses one desirable attribute while being deficient elsewhere.

2.1. A Grid Solution for the Data Problem?

The economic welfare of ethnic minority groups in the UK, raises data issues that require such a multiple data set approach. The basic problem is that non-whites account for a small proportion of the population and sample surveys typically yield minority samples that are too small for meaningful results to be obtained. To some extent the situation is improved when Census data are available. However, while these provide relatively large samples of minority individuals and households, they do not contain any direct measures of income. Other surveys do contain such information but have limited sample sizes when minorities are analysed, with

the problem being especially acute when reporting is required for small area geographies.

A major consequence of such data problems is that important questions about the welfare of minority groups have not been answered. For example, small sample sizes preclude useful measures of household welfare such as poverty rates or inequality measures at anything other than high levels of aggregation. Yet research suggests that disaggregation along two dimensions is crucially important when discussing the welfare of Britain's ethnic minority groups. First, it is clear that treating non-white, minority groups as a homogenous entity is not valid. There is considerable diversity between groups such as Caribbeans, Indians, Pakistanis, Bangladeshis and the Chinese (Leslie, 1998; Modood et al., 1997). This diversity is often quantitatively greater than the differences between non-whites, taken as a whole, and the majority white community. The second dimension where aggregation is important is geographical. Britain's ethnic minorities tend to live in co-ethnic clusters, or enclaves, and this clustering has important consequences for economic activity and unemployment (Clark and Drinkwater, 2002).

The Social Science micro-data sets that could be used to address the above problem are not large from an e-Science perspective. They do, however, tend to be messy and difficult to work with. This problem is compounded for repeated samples (variable definitions change), longitudinal samples (records require information from previous waves), and for the data combination approach considered in this article. Grid technology has the potential to integrate the tasks (data extraction, file transfer) associated with processing quantitative data of this type, by hosting the information in an appropriate manner on a data grid.

2.2. A Grid Solution for the Modelling Problem?

The empirical analysis, a micro-econometric one that relies upon the combination of two or more data sources, belongs to the broader group of modelling techniques associated with linking data sets. Following the terminology of Chesher and Nesheim (2006), who provide a review of this area, as the data linkage is performed with no or unidentifiable common records between the data sets, it falls into the class known as statistical data fusion.

The essence of the approach is to estimate a statistical model on one data set, the so-called donor sample (a relatively small scale but detailed survey), and then apply elements of the fitted model (predicted responses for data imputation, residuals for simulation) to another data set, the so-called recipient sample (possibly larger-scale but less detailed), taking due account of the statistical issues surrounding both model assumptions and data matching as required, such as the potentially heterogeneous nature of the survey data.

Further, the underlying assumptions associated with standard statistical inference may well be violated in a combined analysis. As a consequence, it may be preferable to calculate statistical items such as poverty measure standard errors using re-sampling methods (variants of statistical bootstrapping). As noted by Doornik *et al.* (2004), such analyses have a component that allows for so-called embarrassingly parallelisable computations, and as such are well suited to implementation on the high performance computing (HPC) resources available on the NGS.

3. Methods

The initial plan was to extend an existing macro-econometric application, the SAMD project of Russell *et al.* (2003), to deal with multiple data sources and a different statistical analysis.

3.1. Grid Adoption Issues.

A review of current technologies early in the project resulted in the decision not to re-use software from the SAMD project. Although the broad design principles still applied, technology had moved on significantly since the earlier project was conceived. The intervening years have seen several trends become well established. In particular, Web Service technologies have become widely accepted in the e-Science community, the UK e-Science programme has made a significant investment in OGSA-DAI, and there has been a steady shift towards Web portals to avoid difficulties in deploying complex middleware stacks on end-users' computers. Fortunately, sacrificing re-use of SAMD's redundant technology was offset in part by several factors: the advent of the National Grid Service; the advent of the OGSA-DAI software; and the completion of other projects, which did provide components that we were able to re-use, such as the Athens authorisation software developed for ConvertGrid (Cole *et al*, 2006). After due consideration, the availability and increased maturity of the NGS suggested efforts should be concentrated on their systems, namely Oracle for the data bases and MPI for parallelization.

3.2. The Data Sources

The project has grid enabled two data sources, the British Household Panel Survey (the BHPS) and the 1991 Census Samples of Anonymised Records (the SARs). One can now combine data from the smaller-scale, detailed, BHPS source with the larger sample sizes and geographical coverage of the Census SARs. This lets us provide poverty measures for ethnic minorities which are both broken down by particular ethnic group and geographically disaggregated.

The decision to use the 1991 data needs some comment. This decision was taken when the project was first proposed. The data needs to be readily available to accredited researchers to allow deployment on a data Grid, and it was felt at the time that the confidentiality restrictions envisaged for the so-called Licensed 2001 SARs (the public domain version of the 2001 SARs) would not contain variables suitable for the analysis required. The original release of the Licensed data was not scheduled to contain detailed information on ethnic minorities or on geographical details below regional level. There were also further restrictions (such as the grouping of age) on a variable's response categories. These restrictions are lifted for the Controlled Access (CAMS) version of the 2001 SARs. However, the access and confidentiality constraints imposed on the 2001 CAMS make them presently unusable from a data Grid perspective.

3.3. The Statistical Methodology

This follows an approach in the poverty mapping literature due to Elbers *et al.* (2003). The version of their methodology that is employed is presented here.

3.3.1. Calculations Using the Survey Data, the Donor Sample.

The survey data is used to estimate a model of the economic variable of interest, y_{ic}, which is income in this study. The index i refers to an individual in a sample cluster, which is indexed by c. Each cluster contains n_c observations and there are $N = \sum_{c=1}^{C} n_c$ observations over the C clusters. The economic variable of interest is specified as $\log(y_{ic}) = \boldsymbol{\beta}' \mathbf{x}_{ic} + u_{ic}$ where $\mathbf{x}_{ic}$ is a vector of suitably defined explanatory variables, individual idiosyncratic error terms: $u_{ic} = \eta_c + \varepsilon_{ic}$ where η_c and ε_{ic} are uncorrelated with $\mathbf{x}_{ic}$, independent of each other and IID$(0, \sigma_\eta^2)$ and ID$(0, \sigma_{ic}^2)$ respectively.[1]

First step estimation uses ordinary least squares (OLS) to obtain the coefficient estimates $\hat{\boldsymbol{\beta}}$. The second step uses the fitted residuals. $\hat{u}_{ic} = \log(y_{ic}) - \hat{\boldsymbol{\beta}}' \mathbf{x}_{ic}$, to estimate a model of the variance components. Set[2] $\hat{u}_{ic} = \hat{u}_{.c} + \hat{e}_{ic}$ where $\hat{e}_{ic} = \hat{u}_{ic} - \hat{u}_{.c}$ and proceed to model the idiosyncratic heteroscedastic component σ_{ic}^2 using a logistic style transformed equation

$\frac{\hat{e}_{ic}^2}{(A - \hat{e}_{ic}^2)} = \boldsymbol{\alpha}' z_{ic} + r_{ic}$ where $\mathbf{z}_{ic}$ is a vector of appropriate explanatory variables, A is set to $1.05 \max(\hat{e}_{ic}^2)$ and r_{ic} is a suitable error term.. Estimation of the $\boldsymbol{\alpha}$ coefficients is done using OLS.

Once $\hat{\boldsymbol{\alpha}}$ has been obtained the appropriate prediction for σ_{ic}^2 can be calculated. The remaining variance component, σ_η^2 can be calculated as $\hat{\sigma}_\eta^2 = \max(\hat{V}(\eta_c), 0)$ where

$\hat{V}(\eta_c) = \frac{1}{C-1} \sum_c (\hat{u}_{.ic} - \hat{u}_{..})^2 - \frac{1}{C} \sum_c V(\hat{e}_{.c})$

and $V(\hat{e}_{.c}) = \frac{1}{(n_c - 1) n_c} \sum_{i=1}^{n_c} \hat{e}_{.c}^{\,2}$.

3.3.2. Calculations Using the Census Data, the Recipient Sample.

Once the above estimates have been obtained one can impute the economic variable of interest for any given set of comparable explanatory variables $\mathbf{x}_{kc}$ and $\mathbf{z}_{kc}$. The index k indicates an individual in the Census data source. The Census based predictions can then be calculated, under an assumption of Normality, as: $\hat{y}_{kc} = \exp(\hat{\boldsymbol{\beta}}' \mathbf{x}_{kc} + \frac{\hat{\sigma}_\eta^2 + \hat{\sigma}_{kc}^2}{2})$. The variance prediction $\hat{\sigma}_{kc}^2$ is calculated using $\hat{\boldsymbol{\alpha}}' \mathbf{z}_{kc}$.

One can also calculate a wide variety of poverty measures. The demonstrator uses the parametric (expected) head count (PHC), simulated head count (SHC), and simulated poverty gap (SPG). The PHC measure requires an assumption of Normality and is calculated as

$$\mathrm{PHC}(p) = \frac{1}{n} \sum_{k=1}^{n} \Phi\left((\log(p) - \hat{\boldsymbol{\beta}}' \mathbf{x}_{kc}) \Big/ \sqrt{\hat{\sigma}_\eta^2 + \hat{\sigma}_{kc}^2} \right)$$

where $\Phi(.)$ is the standard Normal distribution function and p is a so-called poverty line. Summation is taken over the sub-sample of interest, ethnic group within geographic area.

If the parametric assumption of Normality was incorrect, this would cause misspecification problems that might affect the estimated measures and their associated standard errors. Simulation can be used to counter this possibility and is used for SHC and SPG. The SHC measure is obtained as the average of B simulated head count measures:

$$\mathrm{SHC(p)} = \frac{1}{B}\sum_{b=1}^{B}\mathrm{SHC(p)_b} \quad \text{where}$$

$$\mathrm{SHC(p)_b} = \frac{1}{n}\sum_{k=1}^{n} I(\,(\hat{\boldsymbol{\beta}}_b'\mathbf{x}_{kc} + \tilde{u}_{k,b} + \tilde{e}_{k,b}\hat{\sigma}_{kc,b}) < \log(p))$$

Note that I(.) is an indicator function taking the value of 1 if the condition inside the parentheses is satisfied and zero otherwise. The standard error predictor, $\hat{\sigma}_{kc,b}$, is calculated using $\hat{\boldsymbol{\alpha}}_b'\mathbf{z}_{kc}$. The coefficients, $\hat{\boldsymbol{\beta}}_b$ and $\hat{\boldsymbol{\alpha}}_b$, are obtained from the b^{th} casewise resample of the survey data, error terms, $\tilde{u}_{k,b}$ & $\tilde{e}_{k,b}$, are drawn with replacement from the error vectors $\tilde{\mathbf{u}}_b$ & $\tilde{\mathbf{e}}_b$.[3]

The SPG measure is obtained in a similar manner:

$$\mathrm{SPG(p)} = \frac{1}{B}\sum_{b=1}^{B}\mathrm{SPG(p)_b}$$

where $\mathrm{SPG(p)_b} = \frac{1}{n}\sum_{k=1}^{n} I(\,\hat{y}_{kc,b} < p) * \left(1 - \frac{\hat{y}_{kc,b}}{p}\right)$.

Here $\hat{y}_{kc,b} = \exp(\hat{\boldsymbol{\beta}}_b'\mathbf{x}_{kc} + \tilde{u}_{k,b} + \tilde{e}_{k,b}\hat{\sigma}_{kc,b})$.

Standard errors are calculated in the usual fashion using the B simulated values of the chosen poverty measure: $\mathrm{SPG(p)_b}$ or $\mathrm{SHC(p)_b}$. A simulated version of the PHC(p), $\mathrm{PHC(p)_b}$ is used to calculate its standard error. This is based upon the PHC(p) equation above, but with the $\hat{\sigma}_\eta^2, \hat{\sigma}_{jc}^2, \hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\alpha}}$ replaced by the values obtained from the b^{th} simulation. Elbers *et al.* (2003) suggest B can be set to 300.

4. The Grid Implementation

The demonstrator is designed for a researcher who wishes to investigate the welfare of ethnic minority groups in the UK. Specifically it allows the researcher to choose different ethnic groups, to specify a level of geography, and to pick from a limited set of poverty measures. The aim is to provide an easy-to-use (Web-based) interface which allows the user to make choices about the type of analysis to be performed and which then returns the results of that analysis to the user. The details of the actual analysis and associated data management are invisible to the user.

The demonstrator service presently produces poverty measures using individual level data. Demonstrator options are the standard headcount measure, and the poverty gap measure. These can be calculated for two possible poverty lines, either 60% of the UK median income or 50% of the UK mean income. The poverty measures are then displayed on a GIS (Geographic Information System) style choropleth map display for UK regional and SARs area geographies. The display presents the poverty measure for the chosen ethnic minority group. The use of individual income data also allows calculation of poverty measures by gender, and, if this option has been chosen for the analysis, the resulting poverty measures can be displayed by ethnic group within gender. The display also produces a box-whisker style plot of the predicted income quantiles[4] for all the ethnic groups associated with a region. This is available for the whole of the UK and at the level of geography displayed by the map (UK region or SARs area).

Other information, such as standard errors, and the results of the model fitted using the survey data, are available from the supporting files returned by the demonstrator. It is also possible to request classification based upon the pseudo-geography available for the 1991 SARs, although these results are not accessible via the visualization tool. The visualisation applet will not display information at a geographic level if the sample size for an ethnic minority group is deemed small. This mimics the type of confidentially restrictions applied to the equivalent controlled access 2001 data.

4.1.1. The Web Service Client

The GEMEDA service stands at the heart of the architecture illustrated in Figure 1. It provides the services and generates the HTML sent to the light client (Web browser) through the handling of events (control). The Spring framework container enforces the MVC (Model View Control) design pattern, enforcing the separation of logic from content and greatly encouraging code re-use.

When a user first accesses GEMEDA he/she creates a user account. An Athens username/password combination is required which is verified on the fly by calling an XML-RPC Athens security interface.

The service downloads a proxy credential automatically through a MyProxy server during each step of the workflow. The proxy credential

lasts 24 hours. The longevity of the Proxy credential is verifiable at all times by clicking on the appropriate link. Connection to the front-end (Web client) is through an HTTPS connection.

The service generates separate OGSA-DAI SQL queries targeted at the SARs and BHPS datasets as a result of user input, and uploads necessary executables to the user's selected HPC computation node if they are not already present.

4.1.2. Security

A user needs login permissions for the GEMEDA service, an e-science certificate to access the NGS, and an Athens username to allow use of the SARs and BHPS data. The user is asked for his/her pass-phrase which is sent using HTTPS to the service. This automatically initiates the creation of a proxy certificate by calling a designated MyProxy server containing the user's certificate. Note that the user's certificate needs to have been uploaded to this server by an appropriate tool. The proxy credential is stored and used by the GEMEDA service throughout the lifetime of the session. A single sign-on mechanism allows the web service to query data through OGSA-DAI and to communicate with the HPC by means of the GSI (Grid Security Infrastructure). This provides message level encryption as well as authenticating and authorising the owner of the proxy credentials.

4.1.3. Grid-enabled Datasets

The SARs and BHPS data sets are stored in separate Oracle databases which occupy slightly over 1 gigabyte of storage space. Data access is done entirely through the OGSA-DAI grid middleware. The Oracle server hosting the SARs and BHPS datasets is a NGS resource administered by the University of Manchester. It should be noted that all the available waves of the BHPS were grid-enabled, along with both the individual and household SARs for 1991. Only the 1991 BHPS wave and individual SARs file are used in the present version of the demonstrator.

Using OGSA-DAI version 4 has proved to be unreliable when accessed securely with GSI. Hence, a local instance of OGSA-DAI accessing the NGS hosted datasets was installed on the Linux server hosting the GEMEDA service. OGSA-DAI query results (XML asynchronous data streams) are uploaded to an FTP server before being converted to a data format recognized by the GEMEDA logic. Converted data sources are then uploaded to the HPC node using secure GridFTP.

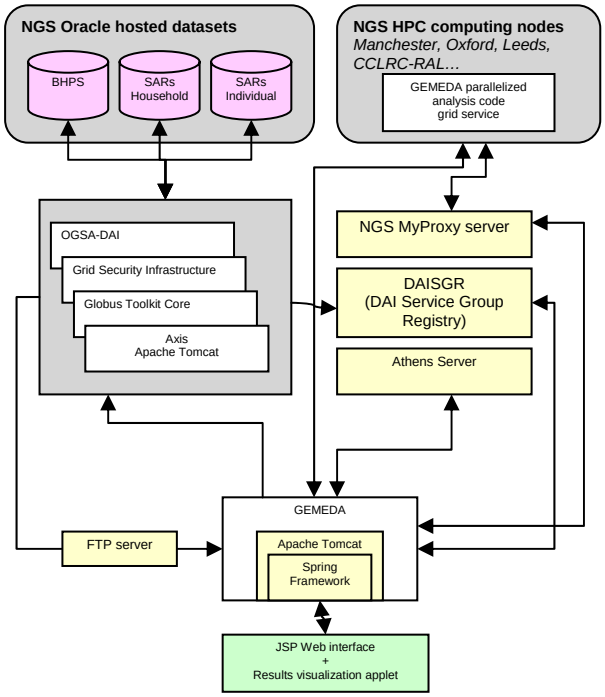


Figure 1: The GEMEDA Architecture

4.1.4. The Analysis Code

The GEMEDA business logic, i.e. the statistical/econometric analysis component, consists of four steps: command/data input, estimation & prediction, welfare measure calculation, and output of the results. These were developed in MPI-parallelized Fortran 95 and produce: model parameter and standard error estimates for the Survey data (the donor sample), imputed income quantiles, poverty measures and associated standard error estimates for the Census data (the recipient sample).

The GEMEDA service accesses the logic modules as a GT 2.4 service through the Globus CogKit running in a separate Java instance (to circumvent namespace clash with the Globus toolkit called by the OGSA-DAI client toolkit). Seen from the level of the GEMEDA Logic, access to the command file, BHPS & SARs data consists in simply reading the appropriate files. ALL files are made accessible to the GEMEDA service code through the GridFTP server.

Event notification is carried by the encapsulating grid service which periodically interrogates the status of the grid service (this does not return a percentage of completion but a status: running, pending, halted, etc).

Once the job is completed, the results are written to file and downloaded through GridFTP by the web service. These results are then converted to XML and spatially mapped by the GEMEDA service before being sent back to the user interface for viewing.

4.1.5. Visualisation.

Once the results files have been returned to the GEMEDA service, they may be viewed in the raw or processed by the service's visualisation applet. A C utility converts the raw data into a form (dbf) appropriate for the applet. This is done for both the regional and SARs area geographies. The applet, which runs under Java 1.4 and above, uses this information along with special mapping data files (shp files) to produce choropleth maps at regional and SARs area geographies for the selected ethnic group and gender category. The shp files are obtained from http://edina.ac.uk/ukborders/ and the applet combines these to produce the map, along with the legend, linked plot, and buttons for gender/ethnic group/geography selection. The applet uses the GeoTools java library 2 to aid the reading and display of the information provided. GeoTools is open source, and was also used by the Hydra I Grid project (Birkin *et al*, 2005). The data and maps remain on the GEMEDA service's server, and permission to use the mapping information is allowed if a user has Athens authentication.

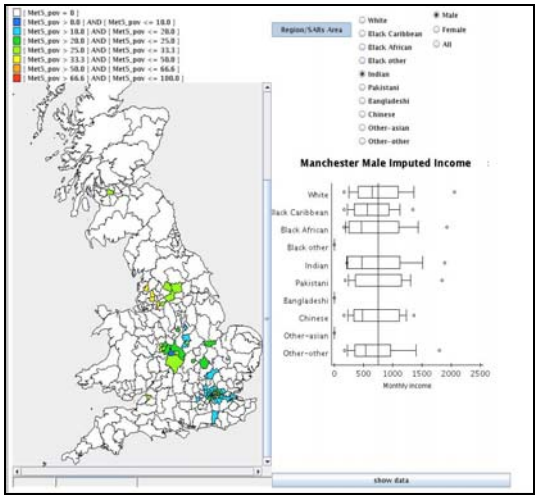


Figure 2: **A Screenshot of the Visualization Applet.**

The poverty measure displayed using the map's colour scheme is the one chosen by the user at job initiation. The applet allows the user to switch between different linked box-whisker style plots of predicted income quantiles by using the map cursor to point to the geographic area of interest. The map has a zoom facility which is useful when the finer SARs area geographies are displayed. Figure 2 presents a screenshot of the SARs area map for Indian Male simulated head count poverty measures using the half mean income poverty line. The linked plot displays predicted income quantiles for all the groups available from the SARs area last pointed to, which was Manchester in this case.

5. A Summary of the Substantive Analysis.[5]

Using data from 1991, models of individual income were estimated using the BHPS data separately for males and females. The specification of the regression equation included all of the variables available via the demonstrator.[6] The results supported splitting the sample by gender as the signs of the parameters on some of the regressors were different for males and females (e.g. the variables indicating marital status and the presence of children in the household). Full regression results are not presented here; instead we note that the explanatory power of both prediction equations appeared reasonable, 53% and 40% for males and females respectively, though the functional form tests suggest there may be room for improvement in the female equation specification. Heteroscedasticity tests strongly rejected the null of homoscedasticity This heterogeneity in the variance was modelled using the methodology of Elbers *et al.* (2003) described in section 3.3 above

The breakdown of poverty measures by region and ethnic group for males shows considerable diversity across each of these dimensions. In general non-whites have higher poverty measures than Whites and this conforms to what we know about the higher unemployment rates and lower earnings of ethnic groups in the UK. Some groups, particularly Black Africans, Pakistanis and Bangladeshis, do particularly badly while the Indians and, to a lesser extent, the Chinese have poverty rates closer to those of Whites. This broad ranking is similar to that in Berthoud (1998). 'Southern' areas of the country generally have lower poverty headcounts than other regions although it should be noted that we do not correct for regional price differentials here. Some regions have relatively high poverty rates for particular groups, for example Bangladeshis and Pakistanis in the North, Pakistanis in the East Midlands and Black Africans in Yorkshire and Humberside, the

West Midlands, the North West and especially Wales.

The broad ranking of the groups and regions is similar for females but poverty rates are much higher. This is because these measures are based on individual income and females have lower participation rates in the labour market. Clearly this measure does not take account of intra-household income transfers. Pakistani and Bangladeshi females stand out as having extremely high poverty rates while, again, Indian and Chinese women are more comparable with their White counterparts.

An advantage of using SARs data is the ability to examine sub-regional geography. However at this level, small samples become a problem. While poverty rates can be estimated for Whites in all the areas, meaningful comparisons between different ethnic groups are only possible for urban areas in which ethnic minorities tend to cluster. There are a number of solutions to the problem of small samples. First, more detail is available using the 2001 Individual SARs which feature considerably more ethnic minority respondents than the 1991 data set. Alternatively, it is possible to capture something of the results for different 'types' of area. Tables 1 and 2 do this using the 'GB Profiles' area classifications attached to the 1991 SARs.[7] All GB profiles that indicate categorisation based on one or more ethnic minorities are grouped together as *Enclaves.* Clark and Drinkwater (2002) suggest that enclaves are associated with worse outcomes for ethnic minorities. The remainder are split into *Poor* (based on housing tenure categorisation), and *The Rest*. The results do not indicate that there is a strong difference in ethnic minority wealth when comparing the SHCs in the *Enclave* profile grouping with the *Poor* profile grouping, however both do worse than *The Rest.*[8]

UK Male.	Profile		
Ethnicity	**Enclave**	**Poor**	**The Rest**
White	23 (1.0)	25 (0.6)	17 (0.4)
Black Caribbean	27 (1.5)	33 (1.6)	22 (1.1)
Black African	39 (2.2)	39 (2.5)	31 (2.0)
Indian	24 (1.3)	25 (1.4)	20 (0.9)
Pakistani	36 (1.6)	31 (1.7)	29 (1.4)
Bangladeshi	37 (2.9)	38 (3.7)	25 (2.1)
Chinese	39 (2.7)	30 (2.2)	25 (1.3)

Table 1: Male SHC Poverty Measures for Profiled Areas.[9]

UK Female.	Profile		
Ethnicity	**Enclave**	**Poor**	**The Rest**
White	42 (1.4)	54 (0.7)	49 (0.6)
Black Caribbean	39 (1.8)	46 (2.0)	39 (1.7)
Black African	47 (2.3)	51 (3.4)	48 (2.5)
Indian	57 (1.6)	56 (2.3)	51 (1.4)
Pakistani	73 (1.8)	72 (2.3)	68 (1.9)
Bangladeshi	71 (2.7)	72 (3.9)	73 (3.1)
Chinese	48 (2.7)	56 (3.3)	48 (2.0)

Table 2: Female SHC Poverty Measures for Profiled Areas.[10]

6. Concluding Comments

The NGS was used to aid the investigation of the welfare of ethnic minorities in the UK by grid enabling the required statistical analysis. This is a small scale problem compared to some science based applications, however, given the local level of resourcing available within the Social Sciences it would not have been possible to obtain the full benefits of a grid implementation without using the NGS.

Our project service arranges for the appropriate data sub-sets to be extracted in a coherent and consistent manner by running queries against OGSA-DAI enabled Oracle databases hosted on the NGS. It then transfers these sub-sets, along with the MPI parallelized code for the statistical analysis, and its associated command file, to a compute node on the NGS. The results of the statistical analysis are then returned to the service, and processed for presentation to the user via a GIS style visualization tool. Job initiation and results viewing are all performed on a web browser.

While the application to ethnic minorities addresses substantive research questions which cannot be addressed adequately using existing techniques, it should be noted that the methodology is general and its future development offers opportunities to social science researchers to address a wide variety of questions using a number of different, complementary data sets. Indeed, the use of OGSA-DAI now offers the potential to include data sets that are hosted as SQL Server databases. In the context of the present analysis, this could be further concurrent data sets, or later versions (2001 for instance) of the BHPS and Census SARs. Extensibility is not restricted to classical quantitative applications, however, and there exists the possibility of integrating appropriate qualitative information using the

techniques of Ahmad *et al* (2005), although this is outside the domain of the authors.

As this was a small project, evaluation of our objectives and usability issues have proceeded in a somewhat *ad hoc* manner and are still ongoing. One area of concern is the level of security required to access the service. The steps required for obtaining and processing an e-certificate are quite involved, and this, combined with problems of access caused by institutional firewalls, can be off-putting for potential users in the Social Sciences.

Notwithstanding this, and other issues such as compute resource brokerage, we regard the establishment of a critical mass of **compatible** Grid-enabled datasets and tools as a necessary condition for the success of e-Social Science in the UK, and hope that the GEMEDA service is at least a useful stepping stone towards this goal.

Acknowledgements

Research supported by ESRC grant number RES-149-25-0009, "Grid Enabled Microeconometric Data Analysis".

We benefited from discussion with members of the following projects: SAMD (Celia Russell, Mike Jones), ConvertGrid (Keith Cole), Hydra I Grid (Mark Birkin, Andrew Turner), and with locally based NGS staff (Matt Ford).

Additional support was provided in the form of an allocation of resources on the NGS itself.

References

Ahmad,K., L.Gillam, D.Cheng (2005), Society Grids, *Proceedings of the UK e-Science All Hands Meeting 2005*, EPSRC Sept.2005

Berthoud, R. (1998), The Incomes of Ethnic Minorities, ISER Report 98-1, Colchester: University of Essex, Institute for Social and Economic Research.

Birkin,M., P.Dew, O.McFarland J.Hodrien. (2005), HYDRA: A Prototype Grid-enabled Decision Support System, *Proceedings of the First International Conference on e-Social Science.*

Chesher, A. and L.Nesheim (2006), Review of the Literature on the Statistical Properties of Linked Datasets, *DTI Economics Papers*, Occasional Paper No.3.

Clark, K. and S.Drinkwater, (2002), Enclaves, neighbourhood effects and economic outcomes: Ethnic minorities in England and Wales, *Journal of Population Economics*, **15**, 5-29.

Cole,K. L.Mason, P.Ekin, J.Maclaren (2006), ConvertGrid, *Proceedings of the Second International Conference on e-Social Science.*

Doornik,J., N.Shepherd, D.F.Hendry (2004), *Parallel Computation in Econometrics: A Simplified Approach*, Nuffield Economics Working Paper.

Elbers,C., J.O.Lanjouw and P.Lanjouw (2003), Micro-level Estimation of Poverty and Inequality, *Econometrica*, 355-364.

Leslie, D. (1998), *An Investigation of Racial Disadvantage*, Manchester University Press, Manchester.

Modood, T., R. Berthoud, J. Lakey, J. Nazroo, P. Smith, S. Virdee, and S. Beishon (1997), *Ethnic Minorities in Britain: Diversity and Disadvantage*, Policy Studies Institute, London.

Russell,C., K.Cole, M.A.S.Jones, S.M.Pickles, M.Riding, K.Roy, M.Sensier (2003), Grid Technology for Social Science: The SAMD Project. *IASSIST Quarterly*, 27#4.

Endnotes

1. IID: identically and independently distributed. ID: independently distributed.
2. The "." subscript means that an average has been taken over that index.
3. $\tilde{\mathbf{e}}_b$ needs to be appropriately standardised.
4. Minimum, 10% quantile, 25% quantile, median, 75% quantile, 90% quantile, maximum.
5. Space restrictions preclude presentation of the regression results, plots, and tables at the regional and SARs levels of geography.
6. Constant, Gender, Age, Age squared, Children present, Marital status, Labour force position, Housing tenure, High qualifications, Immigrant, Region.
7. Results are reported to whole percentages, and one decimal place for the standard error.
8. *Enclave* refers to GBprofile codes 5,13,18, 22, 29 and 33. *Poor* refers to GBprofile codes 1,7,17,21,23,35,36,27,43,44,45,49. *The Rest* refers to the remaining GBprofile codes
9. Black-other, Other-asian and Other-other omitted. SHC is the simulated head count. Standard errors are in parentheses.
10. As 9 above.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

GEODE – Sharing Occupational Data Through The Grid

K.L.L. Tan, [1,2], V. Gayle[1], P.S. Lambert[1], R.O. Sinnott[3], K.J. Turner[2]

1. Department of Applied Social Science, University of Stirling
2. Department of Computing Science and Mathematics, University of Stirling
3. National e-Science Centre, University of Glasgow

Abstract

The ESRC funded Grid Enabled Occupational Data Environment (GEODE) project is conceived to facilitate and virtualise occupational data access through a grid environment. Through GEODE it is planned that occupational data users from the social sciences can access curated datasets, share micro datasets, and perform statistical analysis within a secure virtual community. The Michigan Data Documentation Initiative (DDI) is used to annotate the datasets with social science specific metadata to provide for better semantics and indexes. GEODE uses the Globus Toolkit and the Open Grid Service Architecture – Data Access and Integration (OGSA-DAI) software as the Grid middleware to provide data access and integration. Users access and use occupational data sets through a GEODE web portal. This portal interfaces with the Grid infrastructure and provides user-oriented data searches and services. The processing of CAMSIS (Cambridge Social Interaction and Stratification) measures is used as an illustrative example of how GEODE provides services for linking occupational information.

This paper provides an overview of the GEODE work and lessons learned in applying Grid technologies in this domain.

1. Introduction

1.1 Current occupational data utilisation

Social science surveys are analysed to understand the trend in societies and provide statistics for policy making and planning. Many analyses performed in social science often include occupation as a significant variable.

Occupational information is regularly collected by social science researchers and is usually analysed, or supplied to others, in the form of small electronic datasets, which typically detail occupational unit groups (OUG's). However social researchers are often unaware of how to use such data in an efficient and scientifically consistent way. In particular, occupational data is often analysed and released without documentation, which subsequently raises the barrier for other research efforts [1][15]. Publication of occupational datasets is commonly established via web links, furnished with usage instructions (although the data may be represented in other formats and media for instance email, disc and tape archive). There are also no formal semantic annotations used to define the datasets. Doing this can provide substantial benefits for data searches and access. The current trend is to have descriptions in natural language, which can be ambiguous, within materials accompanying the data sources supplied to end-users. Many of these data resources are also not indexed and therefore do not experience good exposure within the community.

Table A1 describes the existing format of occupational information datasets which have thus far been considered within the project. There are many further occupational information resources in use within the social science research community.

The Grid defines a scalable architecture where data and computational resources are virtualised, abstracted, and collaborated on within virtual organisations [2]. It is therefore highly desirable that occupational data utilisation be made possible in a Grid environment to overcome present issues. This paper illustrates how both the suppliers of occupational information datasets, and the social researchers who may wish to access this data can benefit from a Grid infrastructure developed in the GEODE project [3].

1.2 CAMSIS

CAMSIS scales are measures used by social researchers which indicate the average level of

advantage associated with different occupational unit group positions. CAMSIS scale scores are one of a number of alternative measures of occupational position which are widely used in this field. They are calculated on the basis of a statistical analysis of patterns of social interaction exhibited between individuals from different occupational unit groups.

The use of CAMSIS scales by social researchers illustrates a typical practical scenario of the current practices of distribution and utilisation of occupational data described in Section 1.1. CAMSIS scales are downloaded from a web link with usage instructions put up as descriptions in web pages.

The CAMSIS project [9] is coordinated by members of the GEODE research team so is used as the focal point of initial developments with the service.

1.3 Structure of paper

The paper discusses the GEODE project's intention to improve the practice of occupational data distribution, utilisation, and linking occupational information to CAMSIS scale scores. It presents the requirements and approaches of GEODE in Section 2. Section 3 illustrates the design and the architecture of GEODE and how it is influenced by the application requirements and current technical capabilities of Grid middleware. The results of the development work are discussed in Section 4.

2. Purpose of GEODE

2.1 Objectives and requirements

GEODE [3] aims to improve the current utilisation of occupational data by using the Grid. The goal is to create a virtual community where data resources are virtualised, indexed, and are uniformly accessed by users in a secure manner resulting in a gateway where occupational information can be discovered, exchanged and collaborated on. Occupational data analysis services are rendered to the community members. Occupational data researchers who are the members of this virtual organisation can have their occupational data resources abstracted, described and made accessible in a grid environment, thus standardising the practice of publishing quality datasets.

The GEODE project aims to deliver a usable application that is highly accessible for the users, most of whom have limited prior exposure to Grid services, or to formalised standards of data indexing. The choice is naturally a web interface (because of the ubiquity of web access) representing the view of the Grid, as further discussed in Section 2.4.

The project is also investigating the feasibility of extending its application scope to incorporate other forms of social science datasets in addition to occupational data.

2.2 Occupational data community

An occupational data virtual organisation should encompass disparate data resources made accessible to social science researchers belonging to the community. This is achieved using the MDS (Monitoring and Discovery System) [11], provided by the Globus Toolkit, which provides data aggregation and notification services. The organisation should have fine grained control over user access and the security of the data resources. An indexing service is to be deployed to hold registry information on resources and services. Resources register with the organisation through the indexing service, where resource sharing is then made possible. This is elaborated in the following subsection.

Services make themselves known to the community very much in the same manner as through registration with the index service. The difference is in the metadata used to register with the index service.

2.3 Virtualisation of data resources

The GEODE infrastructure leverages data abstraction Grid middleware (OGSA-DAI [4]) to create a framework for dynamically deploying data resources. The OGSA-DAI middleware, in addition to being able to automatically perform registration with the indexing service, contains the provisions to register custom metadata together with the database schemas.

The Michigan Data Documentation Initiative (DDI) [5] defines a set of XML schemas for annotating social science datasets, thereby promoting the semantic description of the data. The occupational data in the GEODE community is also annotated with social science (custom) metadata (DDI) to give it semantic definitions that are used for yielding more accurate searches than using keywords. The semantic metadata are registered in the community index service when the related data resource is added to the GEODE gateway.

2.4 Usability and accessibility

Many social science researchers are unfamiliar with advanced computer applications. Therefore it is desirable to develop GEODE as an application that can be used with minimal learning and configuration. Though a custom application has been considered, a web portal is much more appealing to the users.

GEODE has developed a web portal as the user interface by which occupational data researchers interact with the grid infrastructure. Through the portal, users can administer their data resources, search the data index, and make requests for statistical services. The portal is accessible via the Internet using standard web browsers. Application users are not bound to specific machines and software in order to perform tasks. This will greatly increase usability in the social science community. The portal was developed with the GridSphere Portal Framework [6], an open-source and widely used tool for portal development.

2.5 Services

The Grid-specific services are built with Globus Toolkit 4 [7]. This WSRF [8] implementation was recently accepted as an OASIS standard. At present, GEODE has developed specific services developed to make queries to the index service, and to link occupational data to CAMSIS scale scores [9]. As the scope of GEODE evolves, services can be readily implemented and deployed on the Grid and accessed via the portal drawing on the service-oriented characteristic of Grid services.

2.6 Security

Security is a major concern, especially when sharing data in the virtual community. Globus Toolkit uses GSI [10] to establish security in a Grid environment. GSI offers authentication, authorization, credential delegation, and single sign-on that GEODE leverages to administer resources, trust and portal service operations. OGSA-DAI makes it possible for resource security to be configured when deployed.

Users delegate their credentials (proxy certificates) to the GEODE services to allow operations to be carried out on their behalf and accounted for.

2.7 Framework Extensibility

Finally, it is highly desirable to make the infrastructure capable of incorporating social data about aspects other than occupation. This benefits social science researchers in other fields, whilst maintaining the same framework whose scope can be extended to provide data and services for more users. Therefore a generic design of the GEODE infrastructure is required to adapt to non-occupational social data.

3. Architecture

3.1 Overview

The high level architectural design of GEODE is depicted in Figure 1. The dotted box represents the occupational data virtual community that comprises the index service, various data services (G1, G2, O4) and application services (in this case the CAMSIS linking service). Users interact with the grid indirectly through the GEODE portal as their web interface. The individual components and functions are elaborated in detail in this section and in section 3.2.

3.1.1 Index service

The GEODE Index Service is considered to be the main core of the virtual community, as services and resources are first discovered prior to performing the actual operations. It is deployed on the default index service supplied by GT4. This index aggregates registration information from both data and application services, where the respective service metadata is propagated by each registering service. This design is not subject to a single index, though currently one instance is deemed sufficient.

3.1.2 Data services

Based on OGSA-DAI configurable data services and appropriate drivers, the GEODE data services provide the data abstraction on occupational data to overcome issues of heterogeneity of data sets including relational tables and text files (comma separated). Data resources are deployed dynamically and register with the index service where discovery could be made. O1, O2 and O3 are examples of the data resources abstracted by the respective data services. GEODE maintains two data services G1 and G2 (as shown in Figure 1). G1 virtualises data that are curated at Stirling locally, and G2 is a collection of resources harnessed from a wider international community of social scientists. The resources of G1 and G2 feature the occupational information described in Table A1.

OGSA-DAI provides a framework for automatic derivation of data access metadata, with provisions made for custom metadata also.

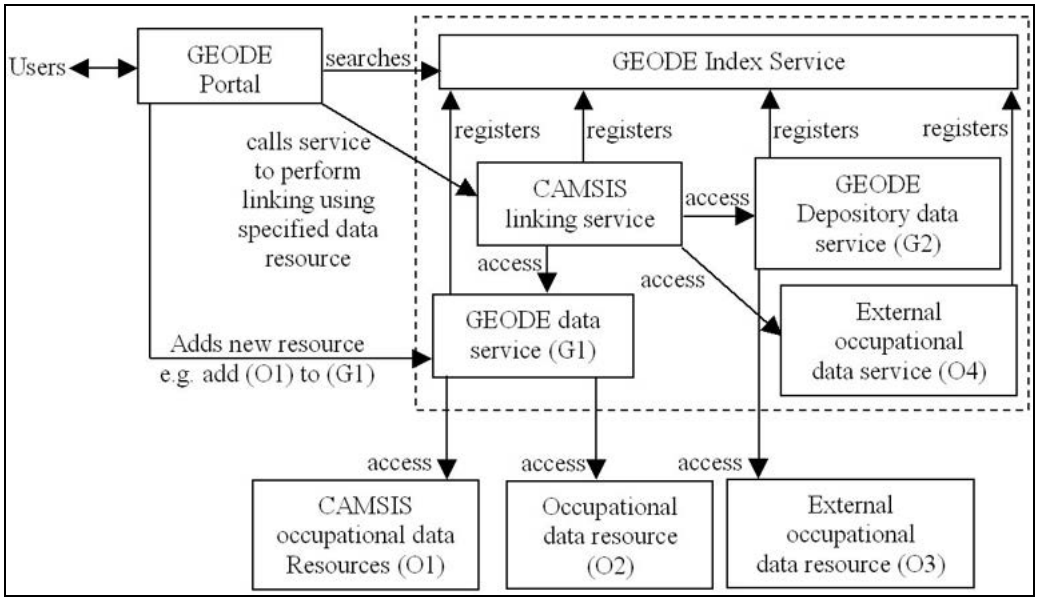


Fig. 1. GEODE Architecture

The DDI schema is used to annotate the data resource with social science metadata which upon registration is aggregated within the index service alongside with the data access metadata.

The GEODE architecture is scalable. An arbitrary number of data services can be deployed (intuitively on different machines/sites) to handle the vast amount of virtualised datasets, both internal and external to GEODE. Users may choose to provide data using their own OGSA-DAI data services (depicted as O4 in Figure 1), considered as external services that register with the index to be made available to all. There can be an arbitrary number of external data services like O4 registering to the GEODE grid, but for illustration purposes only one (O4) is shown.

Users do not interact with the data services directly, but rather do so via the application services which in turn are invoked from the portal.

3.1.3 Application services

The application services are grid services developed using GT4 to provide specific data analysis functions on the data resources. The application services access the data resources and render specified results to the user via the portal (user interface).

Authorization will be verified prior to accessing the services. This is where the GSI is involved to set up a security context with the users and to perform operations on their behalf (via credential delegation).

At the initial stage, only the CAMSIS score linking service will be implemented. As the Grid services are based in Service-Oriented Architecture (SOA), services can be developed and deployed into the Grid with minimal effort.

3.2 Portal

The GEODE portal provides the web interface to users by which operations and functions are invoked by proxy on the Grid services. GridSphere is used to develop the components that make up the entire portal, namely the presentation view, presentation logic, and the application logic. The view is implemented with JSP and the logic with portlets that controls the presentation flow.

The emphasis is on the application logic, developed as a portlet service, which interfaces with the Grid environment. The portlet service invokes the operations of the Grid services and returns results to the presentation logic. GEODE follows the Model-View-Controller design pattern which the occupational data Grid (model), presentation (view) and portlet service (controller) represent.

3.3 Extensibility

The GEODE architecture is designed to be as generic as possible. One of the most promising benefits is to be able to apply or extend the structure towards other social science statistical

data resources with similar requirements. In a generic context, a data Grid with registration to index services along with implemented services can fulfill the requirements of data sharing and collaboration to a considerable and substantial level. In addition to data abstraction and location transparency, this architecture allows control of services whereby the data provider may have the flexibility to provide data services as well as using the services set up within GEODE.

In principle the GEODE Grid can be extended and used for non-occupational social science data as it is designed generically. For example different DDI metadata can be customised for alternative data resources in instances when social scientists have similar requirements for both the storage and distribution of data. Possible areas of application may include the management of geographical and educational data resources, although the scope of this project does not include such implementations.

3.4 Issues

This section discusses the influences and experiences on the technical implementation.

3.4.1 Resource administration

Though GT4 features resource security, OGSA-DAI has yet to utilise this capability (version 2.1). Therefore a temporary measure of data resource administration is clearly required to manage the resource security. Data resources can be deployed by authorised users onto configurable data services. It is natural that the owners or a list of authorised users can manipulate the state, performing tasks such as undeploying resources.

3.4.2 Operations on resources

OGSA-DAI implements activities in a way that they are all invoked via one single operation. Therefore it is not possible to have different security configurations that GT4 supports for individual operations. Although not critical presently, it may become a growing consideration that will impact the project practically. E.g. an activity to modify of resource metadata may have a requirement of authorisation using with an access control list. This requires activity-level security configuration which OGSA-DAI supports shortly after the submission of this paper.

3.4.3 Credential delegation

There are a few ways to delegating credentials to services with GT4. One way is that a client perform the delegation to the services directly. Alternatively the client can store credentials in a depository, where services are then informed of details to retrieve the credentials in order to have the delegated rights. The latter is more favourable as it does not confine proxy credentials to specific locations to perform delegation.

There are currently 2 implementations of credential depository, namely MyProxy [16] and GT4's DelegationService [17]. MyProxy is only available as a software installation in Unix/Linux flavours. GEODE is implemented under Windows and is preferable keep a single environment for maintainability. DelegationService is a Grid service that provides similar functions. Hence this service can be easily deployed into WSRF containers. GEODE aims to use DelegationService as the proxy credential depository. However the current limitation in using DelegationService is that credentials can only be delegated to services that run within the same service container. In cases where services are deployed in multiple containers, the limitation could be resolved by having a DelegationService deployed in each container. Ideally the DelegationService can be used to perform delegations to services in disparate containers.

4. Results

This section reports the results of the prototype development, the current status and progress of GEODE.

4.1 GEODE Prototype

A basic Grid architecture has been developed and deployed, comprising the indexing service and the G1 data service (shown in Figure 1). The indexing service is a mandatory component to establish the virtual community. G1, being an OGSA-DAI configurable data service behaves similarly to G2 and O4. Therefore the latter two services are not required to be deployed for the prototype development. There are no restrictions to how many G1 services deployed. Relational databases and comma-separated value files (local disc and HTTP access) can be deployed as data resources in GEODE.

Custom OGSA-DAI activities were developed for deploying data resources and modifying metadata. Resources, when deployed, automatically register with the

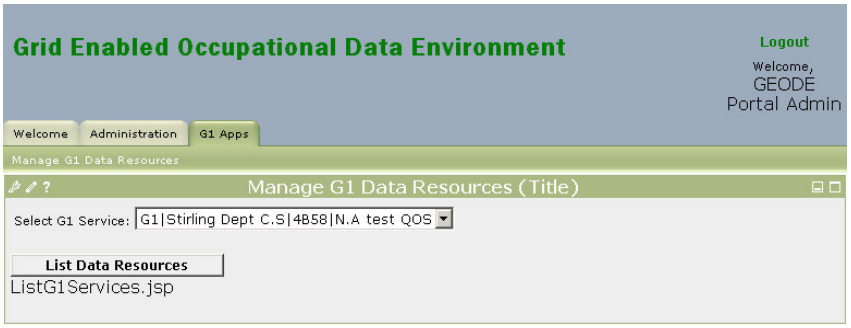


Fig. 2. List of G1 services registered in indexing service

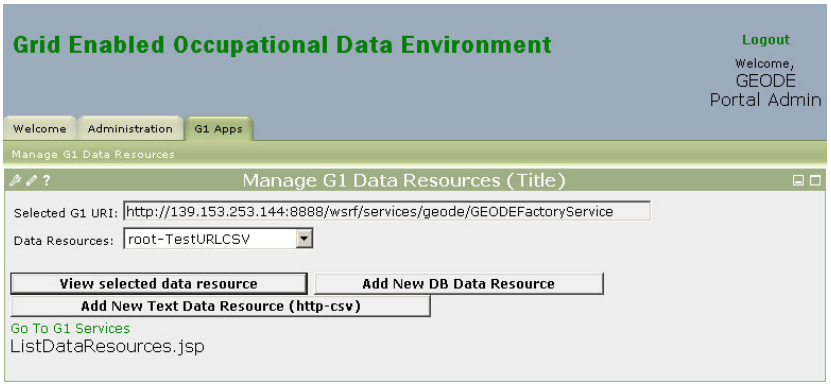


Fig. 3. List of data resources in selected G1 service

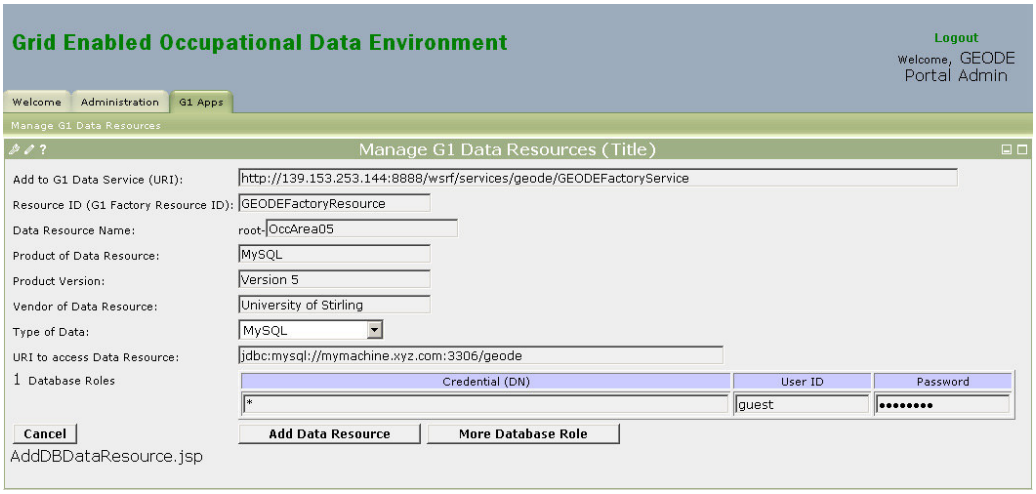


Fig. 4. Add new data resource to selected G1 service

indexing service with the initial specified metadata. The metadata when altered is reflected into the indexing service. Presently the metadata that can be altered is the list of comma-separated value files that are accessible via HTTP. When changed updates to the schema of the data resource that represents it occurs.

The GEODE portal was developed with a user interface that interacts with the indexing service and G1. The views (portlets) accept the user's input, which the portlet services use to communicate with the indexing service and G1. Figures 2 to 5 shows screenshots of the GEODE prototype portal. Figure 2 illustrates the portal querying the indexing service for all G1 services and displays them as a list. The portal

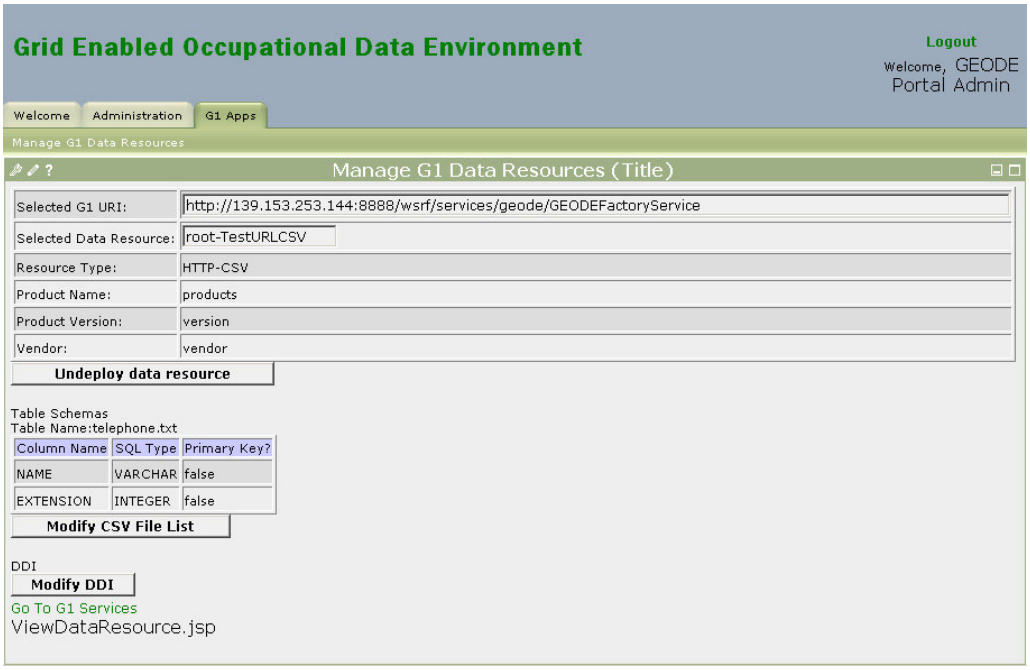


Fig. 5. View data resource in selected G1 service

is also able to list data resources deployed selected G1 services as seen in Figure 3. It is now possible to deploy (see Figure 4) and undeploy (undeploy button in Figure 5) data resources, and to modify the metadata via the portal. In addition, the portal is able to query the indexing service for data resources registered with G1 and to display the metadata of selected data resources. Checks are put in place to guard against the issues listed in Section 3.4.

4.2 Project progress

DDI has been incorporated manually and tested successfully in registering and retrieving from the indexing service. GEODE is currently developing the portal interface and data service activities to manage semantic definitions interactively. The requirements for linking data resources to CAMSIS scores will be examined in detail, and implementation will commence thereafter.

GSI and credential delegation will be assessed and implemented once the functional requirements of GEODE are finalised. To simplify the complexities of client GSI set up, creating proxy certificates and delegating credentials, the web start of the CoG (Commodity Grid) kit [12][13] will be investigated. Java Web Start [14] allows applications to be deployed and launched with a single click from a Web browser, thus omitting complicated and specific installations for GEODE users. This allows researchers to utilise GEODE on other machines instead of being constrained to their own machines.

Prospective users have been identified and would engage in the assessment of using the GEODE prototype when it is ready. This will gather valuable feedback to help in the development of a useful GEODE portal.

5. Conclusion

5.1 Benefits of GEODE

The prototype has illustrated that the design and implementation provides a viable and scalable framework for GEODE and the community members. GEODE and its design in principle can be extended and applied for non-occupational social science data.

GEODE encourages good occupational data utilisation via the portal for occupational information exchange and services, and having the datasets semantically annotated. Occupational data researchers have a common channel for dataset publication that improves data quality definitions, usability and better publicity. The complexity of accessing CAMSIS stratification scale scores will be greatly reduced as a result of using the CAMSIS data linkage service. Likewise other

occupational information linkage services can be applied to meet other user requirements.

5.2 Future work

The possibilities of extending GEODE are great. For example, researchers can create collaborations as resources which authorise members of the community can use, operate on, and share. Datasets in XML format can be virtualised readily with OGSA-DAI as the middleware, although this is not required in the scope of GEODE. Analytical and often time-consuming statistical services can be developed and deployed to the Grid. There is a substantial user-community in the social sciences who would benefit from utilising the GEODE services. Additionally, GEODE can be extended to incorporate non-occupational data and services. Ideally GEODE can be established as the portal for a wide range of social science data.

Appendix

Table A1: Selected Occupational Information Datasets used in GEODE

1. CAMSIS indexes, www.camsis.stir.ac.uk/versions.html	
Format:	Index matrix, SPSS and plain text data files
Units:	Variety of national OUG, plus gender, employment status
Output:	CAMSIS scale scores
2. CAMSIS occupational information value labels, www.camsis.stir.ac.uk/occunits/distribution.html	
Format:	One-to-one translation, SPSS and plain text files
Units:	Variety of national OUGs
Output:	Text labels to numerical OUG codes
3. ISEI tools, home.fsw.vu.nl/~ganzeboom/pisa/	
Format:	One-to-one translation, SPSS and plain text files
Units:	ISOC-88 and -68 international OUG schemes
Output:	ISEI and SIOPS occupational scale scores
4. E-SEC matrices, www.iser.essex.ac.uk/esec	
Format:	Index matrix, MS-Excel and SPSS syntax
Units:	ISCO-88 OUG, and employment status (es) data
Output:	E-SEC class position for OUG-es combination
5. Hakim gender segregation codes (Hakim, C. 1998 *Social Change and Innovation*, Oxford University Press, pp266-90).	
Format:	One-to-one translation, paper printout
Units:	ISCO-88 and UK SOC90 OUG schemes
Output:	Gender segregation information for OUG codes

References

[1] P.S. Lambert, "Handling Occupational Information", *Building Research Capacity*, pp 4:9-12, Nov. 2002.

[2] I. Foster, C. Kesselman, S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 15(3), 2001.

[3] P.S. Lambert, V. Gayle, K. Prandy, R.O. Sinnott, K.L.L. Tan, K.J. Turner, GEODE Grid Enabled Occupational Data Environment, http://www.geode.stir.ac.uk, Oct. 2005.

[4] OGSA-DAI, Open Grid Service Architecture, Data Access and Integration, http://www.ogsadai.org.uk, Feb 2006.

[5] DDI, Data Documentation Initiative, http://www.icpsr.umich.edu/DDI/, Feb. 2006.

[6] Gridsphere Portal Framework, http://www.gridsphere.org, Dec. 2005

[7] I. Foster, Globus Toolkit Version 4: Software for Service-Oriented Systems, IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2005.

[8] WSRF specification, Web Services Resource Framework (WSRF) v1.2 Specification.

[9] P.S. Lambert and K. Prandy, CAMSIS: Cambridge Social Interaction and Stratification scales, http://www.camsis.stir.ac.uk/, Aug. 2005.

[10] The Globus Alliance, Grid Security Infrastructure, http://www.globus.org/toolkit/docs/4.0/security/key-index.html, Apr. 2006.

[11] Jennifer M. Schopf, Monitoring and Discovery in a Web Services Framework: Functionality and Performance of the Globus Toolkit's MDS4, http://www-unix.mcs.anl.gov/~schopf/Pubs/mds-sc05.pdf, Apr. 2006.

[12] Java CoG Kit – Webstart Applications, http://www.cogkit.org/release/4_1_2/webstart/, Feb. 2006.

[13] Java CoG Kit, http://www.cogkit.org/, Feb. 2006.

[14] Java Web Start Overview White Paper, http://java.sun.com/developer/technicalArticles/WebServices/JWS_2/JWS_White_Paper.pdf, May 2005

[15] P.S. Lambert, K.L.L. Tan, V. Gayle, K. Prandy, R.O. Sinnott, Developing a Grid Enabled Occupational Data Environment, to appear Second International Conference on e-Social Science, Manchester, UK, June 2006.

[16] The Globus Alliance, GT4.0: Credential Management: MyProxy, http://www.globus.org/toolkit/docs/4.0/security/myproxy/, Apr. 2006.

[17] The Globus Alliance, GT4.0: Security: Delegation Service, http://www.globus.org/toolkit/docs/4.0/security/delegation/, Apr. 2006.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

The National Centre for e-Social Science

Rob Procter[1], Mike Batty[2], Mark Birkin[3], Rob Crouchley[4], William H. Dutton[5], Pete Edwards[6], Mike Fraser[7], Peter Halfpenny[8], Yuwei Lin[1] and Tom Rodden[9]

[1]National Centre for e-Social Science, University of Manchester
[2]GeoVUE, Centre for Advanced Spatial Analysis, UCL
[3]MoSeS, School of Geography, University of Leeds
[4]CQeSS, Centre for e-Science, University of Lancaster
[5]OeSS, Oxford Internet Institute, University of Oxford
[6]PolicyGrid, Department of Computing Science, University of Aberdeen
[7]MiMeG, Department of Computer Science, University of Bristol
[8] National Centre for e-Social Science, University of Manchester
[9]DReSS, Department of Computer Science, Nottingham University

Abstract

This paper outlines the work of the UK National Centre for e-Social Science and its plans for facilitating the take-up of Grid infrastructure and tools within the social science research community. It describes the kinds of social science research challenges to which Grid technologies and tools have been applied, the contribution that NCeSS can make to support the wider take-up of e-Science and, finally, its plans for future work.

1. Introduction

The concept of e-Science refers to the growing use of advanced Internet and Grid applications to support scientific research. The National Centre for e-Social Science (NCeSS) is funded by the Economic and Social Research Council (ESRC) to investigate and promote the use of e-Science to benefit social science research. The ESRC's investment in e-Science began in 2003 with the funding of eleven pilot demonstrator projects (PDPs). The creation of NCeSS followed in 2004. The overall goal of NCeSS is to stimulate the uptake and use of emerging e-Science technologies within the social sciences.

2. NCeSS Overview

The Centre is made up of a co-ordinating Hub based at the University of Manchester with support from the UK Data Archive at the University of Essex, plus seven research Nodes based at institutions throughout the UK. A series of smaller e-Social Science projects have been commissioned under the ESRC small grant scheme.

The Hub co-ordinates the research activities of the Centre as well as providing e-Social Science training, technical support, information services and support to users. The Hub acts as the central resource base for e-Social Science issues and activities in the UK, integrating them with ESRC research methods initiatives and the existing e-Science core programme.

The majority of the Centre's research is undertaken in seven research Nodes, each of which focus on a different area of e-Social Science and funded for three years. There are also twelve Small Grant projects, each funded for one year.

The NCeSS research programme was conceived from the beginning around two distinct strands. The first, the applications strand, is aimed at stimulating the uptake and use by social scientists of new and emerging Grid computing and data infrastructure in order to make advances in both quantitative and qualitative economic and social research. It seeks to draw upon and further advances generic middleware developments from the e-Science core programme and apply them to the particular needs of the social science research community in order to generate new solutions to social science research problems. The second, the social shaping strand, examines the social and economic influences on the development of e-Science and, conversely, the socio-economic impact of Grid technologies. This strand focuses on the factors influencing the design, uptake and use of Grid technologies, and the conditions determining whether and how their potential is realised. As such, the social shaping strand has

potential relevance well beyond the social sciences and we will return to this later.

2.1 Collaboratory for Quantitative e-Social Science (CQeSS)

The overall aim of CQeSS is to ensure the effective development and use of Grid-enabled quantitative methods. Part of the focus of CQeSS is on developing middleware that allows users to exploit Grid resources such as datasets while continuing to employ their favourite desktop analysis tools [6].

As e-Social Science develops, there will be a growing user base of social researchers who are keen to share resources and applications in order to tackle some of the large-scale research challenges that confront us. They will be aware of the potential of e-Science technology to provide collaborative tools and provide access to distributed computing resources and data. However, social scientists are not ideally catered for by the current Grid middleware and often lack the extensive programming skills to use the current infrastructure to the full and to adapt their existing "heritage" applications.

This problem prompted the Grid Technology Group at CCLRC Daresbury Laboratory to write the prototype GROWL: Grid Resources On Workstation Library toolkit.[1] This toolkit provides an easy-to-use client-side interface. CQeSS is now seeking to apply the GROWL middleware to wrap the statistical modeling methods available in SABRE as well as other computationally-demanding models and to make these developments available in the distributed environment as a componentised R library and as a Stata "plug-in".

2.2 New Forms of Digital Record for e-Social Science (DReSS)

DReSS seeks to understand how new forms of record may emerge from and for e-Social Science. Specifically, it seeks to explore the development of new tools for capturing, replaying, and analyzing social science records [3,4,5,12]. Social scientists work in close partnership with computer scientists on three Driver Projects to develop e-Social Science applications demonstrating the salience of new forms of digital record.

Development work within the Driver Projects focuses on the assembly of qualitative records, the structuring of assembled records, and the coupling of qualitative and quantitative records. The research is underpinned by three key themes – record, re-representation, and replay – to ensure the development of services that have some general purchase and utility. Work to date has seen the development of a dedicated ReplayTool[2] which supports the use of records containing multiple forms data from multiple sources, allows data to be represented and re-represented in different ways to support different kinds of analysis, and enable researchers to replay digital records to support analysis. Current work seeks to implement semantic web ontologies to support structured forms of analysis and is also exploring the potential of vision recognition and text mining techniques to support automatic coding of large datasets. The work of the Node is driven by an iterative user-centred prototyping approach to demonstrate the salience of new forms of digital record to future social science research.

2.3 Modelling and Simulation for e-Social Science (MoSeS)

MoSeS will provide a suite of modeling and simulation tools which will be thoroughly grounded in a series of well-defined policy scenarios. The scenarios will be validated by both social scientists and non-academic users [1]. Moses is particularly focused on policy applications within the domains of health care, transport planning and public finance. Social science problems of this type are characterized by a requirement for extensive data integration, the need for multiple iterations of computationally intensive scenarios, and a collaborative approach to problem-solving.

The key objective is to develop a representation of the entire UK population as individuals and households, at present and in the future, together with a package of modeling tools which allows specific research and policy questions to be addressed. The advances it will seek to achieve include the creation of a dynamic, real-time, individually-based demographic forecasting model; for defined policy scenarios, to facilitate integration of data and reporting services, including Geographical Information Systems (GIS), with modeling, forecasting and optimisation tools, based on a secure grid services architecture; to use hybrid agent-based simulations to articulate the connections between individual level and structural change in social systems; and to provide high level capability for the articulation of unique evidence-based user scenarios for social research and policy analysis.

[1] http://www.growl.org.uk

[2] www.ncess.ac.uk/nodes/digitalrecord/replaytool/

2.4 Semantic Grid Tools for Rural Policy Development and Appraisal (PolicyGrid)

PolicyGrid brings together social scientists with interests in rural policy development and appraisal with computer scientists who have experience in Grid and Semantic Web technologies. The core objective is to explore how Semantic Grid tools [10] can support social scientists and policy makers who increasingly use mixed-method approaches combining qualitative and quantitative research techniques (e.g., surveys and interviews, ethnography, case studies, simulations).

Provision of metadata infrastructure in this context (to support annotation and sharing of resources) presents many challenges including the dynamic and contested nature of many concepts within the social sciences, the need to align with existing thesauri (where those exist) and the need to support open, community based efforts. The Node is thus exploring so-called "folktology" solutions which exploit both lightweight ontology and folksonomy (social tagging) based approaches [13]. Another activity is investigating the use of argumentation approaches [14] as a means of facilitating evidence-based policy making; argument structures provide a mechanism for linking qualitative analyses with other resources including simulation experiments, and queries over Grid-enabled data services. Enhanced support for social simulation on the Grid [15] is another focus, with a framework for characterising simulation models under development, as well as workflow support.

2.5 Mixed Media Grid (MiMeG)

MiMeG is developing tools to support distributed, collaborative video analysis [11]. The project arises from converging developments in contemporary social science. Digital video is becoming an invaluable tool for social and cognitive scientists to capture and analyse a wide range of social action and interactions; video-based research is increasingly undertaken by research teams distributed across institutions in the UK, Europe and worldwide; and there is little existing support for remote and real-time discussion and analysis of video data for social scientists. Therefore MiMeG has taken as a central concern how to design tools and technologies to support remote, collaborative and real-time analysis of video materials and related data. In developing these tools the project has the potential to support video-based 'collaboratories' amongst research teams distributed across institutions, disciplines and locations. We are undertaking studies of research and analytic practice to develop requirements for the design of these tools, including detailed qualitative studies of current practice in fields of video-based research in both social scientific and professional communities.

MiMeG has built prototype software that can connect collaborators, allowing them to see, discuss and annotate video together in real-time and from remote locations. This software is now freely available under a GPL license in PC and Mac versions and allows real time analysis of video data between two or more remote sites. The software also enables each site to see the others' notes, transcripts, drawings etc on the video they are watching and talk to each other at the same time. We are now beginning development of a second version of the software, which will begin to consider more innovative ways of capturing and representing communicative conduct in remote data sessions.

MiMeG has collected a wide-ranging data on existing video-based research practice across the social and cognitive sciences and the work of video analysts working in other occupations. Now that the prototype tools are beginning to be adopted we are undertaking a series of studies of their use by social scientists as part of their everyday research activities.

2.6 Geographic Virtual Urban Environments (GeoVUE)

GeoVUE will provide grid-enabled virtual environments within which users are able to link spatial data to geographic information systems-(GIS)-related software relevant to a variety of scientific and design problems. It will provide decision support for a range of users from academics and professionals involved in furthering our geographic understanding of cities to planners and urban designers who require detailed socio-economic data in plan preparation. At the heart of GeoVUE lies the concept of a VUE which represents a particular way of looking at spatial data with respect to the requirements of different kinds of user, thus defining a virtual organisation relevant to the problem in hand.

GeoVUE is developing three demonstrators of increasing complexity and sophistication which will form part of a structured process of sequential development to the ultimate aim of producing a generic framework.

2.7 Oxford e-Social Science Project (OeSS)

OeSS focuses on the inter-related social, institutional, ethical, legal and other issues surrounding e-Science infrastructures and research practices. The design and use of advanced Internet and Grid technologies in the social, natural and computer sciences are likely to reconfigure not only how researchers get and provide data resources and other information but also what they and the public can access and know; not only how they collaborate, but with whom they collaborate; not only what computer-based services they use, but from whom they obtain services [8]. This reconfiguring of access to a wide variety of resources raises numerous issues, including ethical concerns (e.g., confidentiality, anonymity, informed consent), legal uncertainties (e.g., privacy and data protection, liability), social (ownership, trust, credit), and institutional (IPR and risk in multi-institutional collaboration) issues [7,9,16]. OeSS assembled a multi-disciplinary team to analyze e-Sciences in the UK and globally, focusing on a set of in-depth case studies to uncover the dynamics of these ethical, legal and institutional issues that facilitate or constrain the use of e-Science data, tools and other resources, and shape initiatives to address them.

3. Social Shaping

Social shaping is defined very broadly within the NCeSS programme to include all social and economic aspects of the genesis, implementation, use, usability, immediate effects and longer-term impacts of the new technologies. Despite the very substantial current investment in the Grid, little is known about the nature and extent of take-up, about how and why and by whom these new technologies are being adopted, nor what will be their likely effects on the character and conduct of future scientific research, including social scientific research.

While OeSS is the only NCeSS node which has social shaping research as its principal aim, a number of them address this as a subsidiary aim and it is also a focus for a number of the Small Grant projects. The 'entangled data' project (Essex University), for example, has been conducting a comparative study to understand how groups of research scientists collaborate using shared digital data sources, developing insights into the likely use and non-use of e-Science technologies, and the social and technological innovations that may be required as e-Science expands from its early adopters [2].

There are also important questions to be answered about how Grid-based tools might be adapted to make them more usable. Members of NCeSS have been active organising workshops on usability issues for e-Science[3]. Finally, the Hub is working with the UK e-Science Institute to investigate how barriers to the wider take up of Grid technologies can be tackled.

4. Developing e-Social Science

Much of what is presently understood about e-Social Science is an extrapolation of existing practice and is focused around areas where it is believed the Grid can address known limitations of research methods.

Computer-based modeling and simulation is a well established social science research tool. As models get more complex, they need computing power and there are similar benefits to be had from the Grid for statistics-based research generally.

The sharing and re-use of data is already well established in the social sciences, but heterogeneity in formats means that linking different datasets together can still prove very difficult. In partnership with UK data centres, NCeSS has begun pilot studies of 'grid enabling' selected datasets.

The vast amounts of data generated as people go about their daily activities are, as yet, barely exploited for research purposes. For example, use of public services is captured in administrative records; in the private sector, patterns of consumption of goods and services are captured in credit and debit card records; patterns of movement are logged by sensors, such as traffic cameras, satellites and mobile phones; the movement of goods is increasingly tracked by devices such as RFID tags. Exploiting these data sources to their full research potential requires new mechanisms for ensuring secure and confidential access to sensitive data, and new tools for integrating, structuring, visualisation and analysis.

How e-Social Science will develop in the longer term will become clearer as researchers explore and experiment with the opportunities the Grid provides. To facilitate this process, NCeSS has been organizing a series of Agenda

[3] See, for example, www.nesc.ac.uk/action/esi/contribution.cfm?Title=613 and www.ncess.ac.uk/support/chi/index.php/Main_Page

Setting Workshops (ASWs)[4] to which social scientists are invited to hear about opportunities for using the Grid in research and to reflect on how these technologies might address the obstacles. Nine ASWs have been held over the past eighteen months and more are planned.

4.1. Widening Engagement

The ASWs have helped to identify new areas for the application of Grid technologies in the social sciences and community needs. For example, a theme emerging from several of the ASWs is that Grid-enabled datasets, services and tools are key enablers for the wider take-up of e-Social Science. They have also enabled NCeSS to profile the social science research constituency in terms of its awareness and readiness to adopt new technologies. We have used these findings to inform NCeSS strategy for widening the community's engagement with e-Social Science.

Our social science researcher profile identifies three distinct communities – the 'early adopters' who are keen to push to the limit of what is possible; the 'interested' who will adopt new research tools and services if they believe these will provide simple ways of advancing their research; and the 'uncommitted' who have yet to appreciate the relevance of Grid technologies for their research.

Those early adopters who have not been recruited into the NCeSS programme will demand research tools they can apply. Tools are also needed to convert the interested into adopters and demonstrators which will convince the uncommitted to join the ranks of the interested.

To address the needs of these communities, NCeSS has begun to build an e-Infrastructure for social sciences. To engage with the uncommitted, NCeSS will deploy a selection of demonstrators from those being developed within the Nodes (e.g., GeoVUE visualisation tools), NCeSS Small Grants, and demonstrators developed by the e-Social Science PDPs. More generally, NCeSS will provide a platform for disseminating the benefits of e-Social Science to the wider research community, leverage existing e-Social Science and e-Science investments, and ensure the usability and sustainability of middleware, services and tools.

4.2 Solving Real Social Science Problems

The diffusion of any innovation, including e-Social Science tools and practices, will be shaped by the degree it can address the problems of potential users. NCeSS is developing an increasingly concrete understanding of the applications of e-Sciences in the addressing social science problems. The examples illustrate how social scientists can combine its component parts – i.e., datasets, services and tools – in flexible yet powerful ways to overcome various kinds of problems they face in pursuing their research.

For example, solving complex statistical modeling problems often involves multiple steps and iterations where the output of one step is used as the input in the next: 1) select data or subsets, maybe from more than one source; 2) merge, harvest or fuse; 3) input to model and analyse; 4) repeat previous cycle with new or different data; 5) repeat with different model assumptions, parameters and possibly data; 6) synthesise outputs from multiple models or world views; 7) load output into a different analysis or visualisation tool.

Managing these steps manually is potentially difficult, and performing the data integration and modeling using desktop PC tools may be very time consuming. For example, analysis of work histories requires many different sources of data which need to be reconciled and integrated in order to produce a coherent and contextualised life or work history and takes one week on a desk top PC in Stata. To then simultaneously analyse the data could take months on a desktop PC running serial SABRE and over 10 years using Stata. This presents a major obstacle to research. Using Grid-enabled data, analysis and workflow tools, however, the researcher can compose complex sequences of steps using different computational and data resources, and execute them (semi) automatically using powerful computers in a comparatively short time.

Schelling's model of segregation provides a way of exploring how the spatial distribution of society groups ('agents') responds to different assumptions about neighbour preferences. For example, a social scientist interested in racial segregation could use it to explore questions such as:

- Sensitivity of the model to initial arrangements of agents.
- The relationship between ratio of different 'races' and the final segregation ratio.

To do this, the researcher might need to run simulations for 50 different values of the initial race ratio and 100 different initial agents' arrangements. Using a desktop PC, 5000 simulations will take about one week which may well make it impractical. If the researcher

[4] Funded by the ESRC/JISC ReDReSS project.

has access to Grid computational resources, however, the same number of simulations can be run in about five minutes, opening up new possibilities for the researcher.

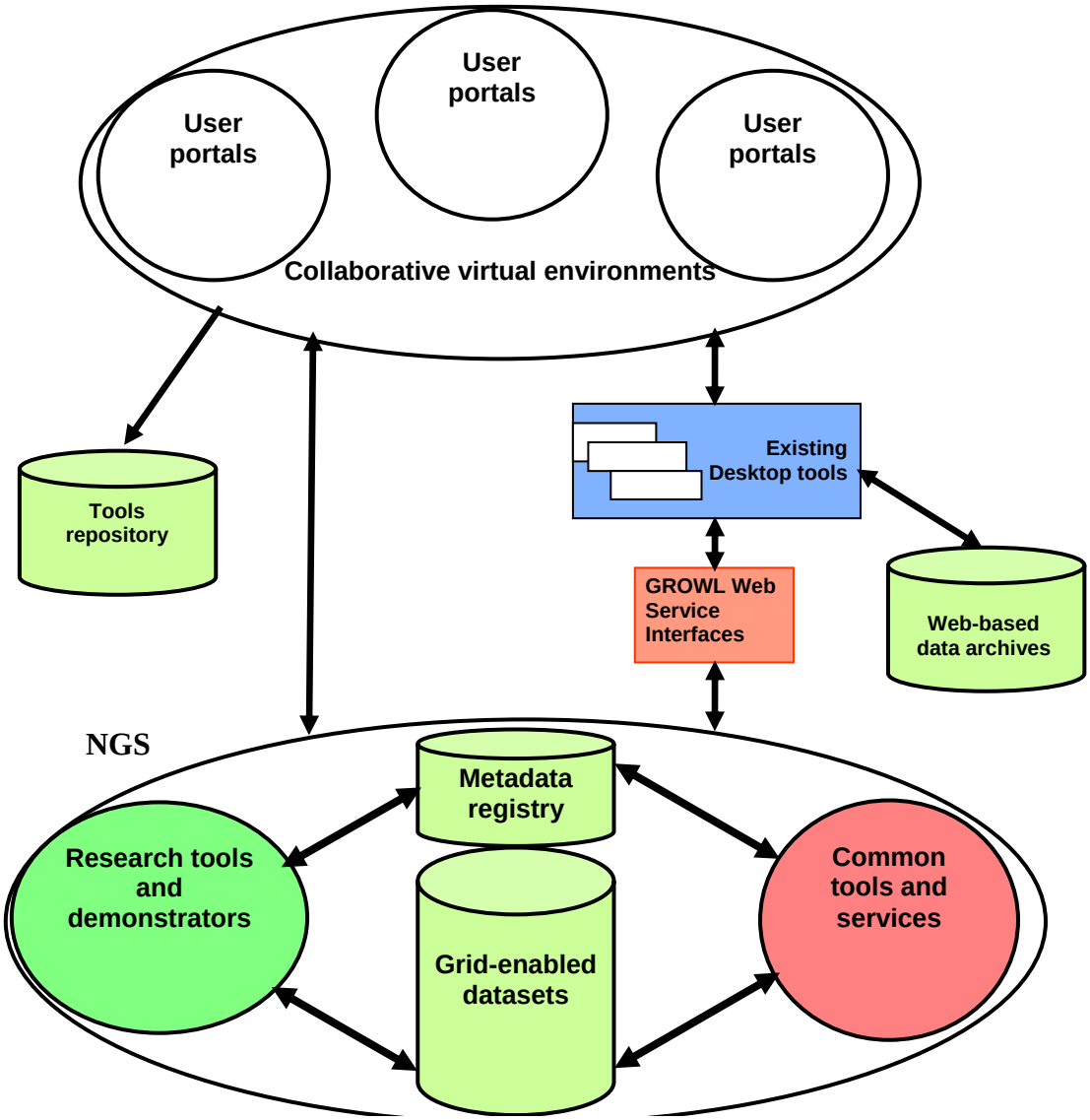


Figure 1: e-Infrastructure for Social Sciences.

5. e-Infrastructure Architecture

An overview of the e-Infrastructure architecture is show in Figure 1.

Portals provide an integrated, single point of access to e-Infrastructure resources through a familiar and simple-to-use web-style user interface which hides the underlying complexity. Users can authenticate themselves, discover resources (data, tools and services) and create their own 'workflows' from tools and services to carry out analysis. A tool or service is mapped to a Java portlet for insertion into one or more portal frameworks. Each Node can deploy its own preferred framework and choose from a repository of portlets depending on its user requirements. NCeSS will also host fully-functional portals with both collaboration tools and service portlets for access to Grid-enabled datasets and tools, including tools for generic tasks such as resource discovery, workflow composition, data mining and visualisation. These portals will be designed to be easily usable by a wide range of users. Collaborative virtual environments (such as Sakai) help distant researchers: share in the research tasks, run project meetings, discuss results, and work up presentations and papers.

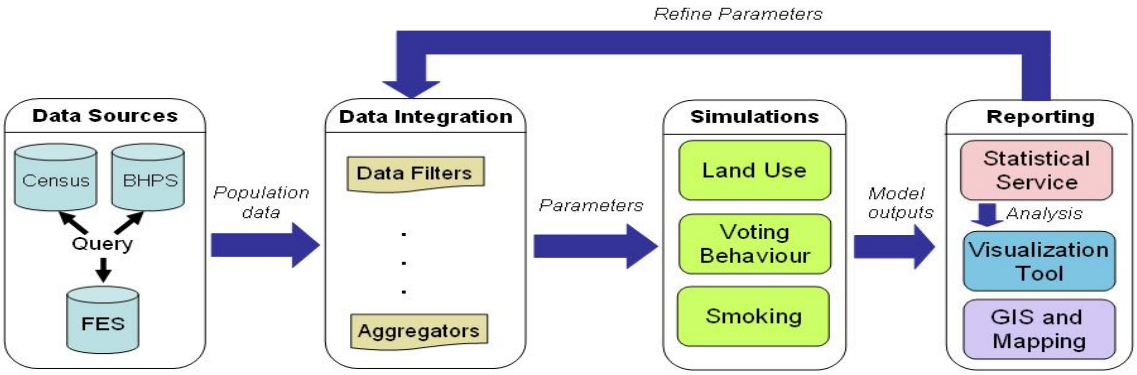


Figure 2: Example simulation workflows.

A range of research tools and demonstrators selected from those being developed within the NCeSS programme will be deployed, with selection criteria informed by ongoing consultations with the wider social science research community. Agenda Setting Workshops have already enabled us to identify simulation and modeling as a priority area. Building on the work of MoSeS and PolicyGrid, simulation and modeling tools will be deployed which will enable researchers to create their own workflows to run their own simulations, visualise and analyse results, and archive them for future comparison and re-use (see Figure 2). This will facilitate development and sharing of social simulation resources within the UK social science community, encourage cooperation between model developers and researchers, and help foster the adoption of simulation as a research method in the social sciences.

A set of common tools and services (e.g., workflow, visualisation) will also be made available. Our approach to e-Infrastructure building does not preclude local community-level deployments of tools and services. Indeed, the NCeSS programme is uncovering requirements for usability, control and trust in qualitative research that may be more appropriately served through localised provision of tools and data. For example, there may be issues with the distribution of sensitive video data to (and via) un-trusted third parties. Development of desktop tools and community-oriented service deployments is part of the ongoing NCeSS e-Infrastructure strategy, particularly in supporting qualitative research. MiMeG and DReSS are building new tools and have a remit within their existing work programmes to capacity build and deploy to interested communities.

A range of Grid-enabled datasets are being made available (quantitative and qualitative), including datasets already Grid-enabled by the PDPs or being Grid-enabled by UK data centres. Others will be selected after consultation with Nodes and Small Grants to identify dataset usage within the NCeSS programme; the major data centres to identify patterns of dataset usage (quantitative and qualitative) within the wider social science community, to understand licensing issues and ensure complementarity with JISC funded Grid-enabling activities (current and planned); and with the social science community to identify research drivers and ensure a fit with the ESRC's future data strategy plans.

6. Benefits for Social Sciences

The e-Infrastructure project will serve a number of important objectives which are relevant to the social science and wider research communities:

- enhance understanding of issues around resource discovery, data access, security and usability by providing a testbed for the development of metadata and service registries, tools for user authorisation and authentication, and user portals;
- lay foundations for an integrated strategy for the future development and support of e-Social Science infrastructure and services;
- leverage the infrastructure investment being made by UK e-Science core programme and JISC for the benefit of the Social Sciences;
- promote synergies across NCeSS and other ESRC investments, co-ordinate activities, encourage mutual support and identify areas in which to promote the benefits of common policies and technology standards.

NCeSS is working closely with the social science community to ensure that the project is driven by research needs and, specifically, to identify the most research-relevant resources, tools and services to incorporate into the e-Infrastructure. It is also working with the UK e-Science core programme (NGS, OMII-UK consortium, NGS, DCC) and JISC to devise an e-Infrastructure development plan which will define technical standards and mechanisms to ensure long term sustainability.

7. Summary and Future Work

In this paper we have presented an overview of how the NCeSS programme is working to develop applications of Grid technologies and tools within the social sciences and to understand the factors which encourage or inhibit their wider diffusion and deployment. As such, NCeSS has lessons for the e-Science community as it begins to grapple with these same problems.

NCeSS will continue to develop its agenda for e-Social Science. Activities focusing on the usability of e-Science are expanding through participation in JISC VRE and EPSRC usability and e-Science programmes. NCeSS is also beginning a series of fieldwork investigations of work practices in developing areas of e-Science so as to better understand the impact of these new technologies and the issues they raise for usability, and the methodologies used for requirements capture, design and development.

Bibliography

1. Birkin, M., Clarke, M., Rees, P., Chen, H., Keen, J. and Xu, J. (2005). MoSeS: Modelling and Simulation for e-Social Science. In Proc 1st Int. Conf. on e-Social Science, Manchester.
2. Carlson, S. and Anderson, B. (2006). e-Nabling Data: Potential impacts on data, methods and expertise. In Proc 2nd Int. Conf. on e-Social Science, Manchester.
3. Crabtree, A. and Rouncefield, M. (2005). Working with text logs: some early experiences of e-SS in the field. In Proc 1st Int. Conf. on e-Social Science, Manchester.
4. Crabtree, A. French, A., Greenhalgh, C., Rodden, T. and Benford, S. (2006). Working with digital records: developing tool support. In Proc 2nd Int. Conf. on e-Social Science, Manchester
5. Crabtree, A., French, A., Greenhalgh, C., Benford, S., Cheverst, K., Fitton, D., Rouncefield, M. and Graham, C. (2006). Developing digital records: early experiences of record and replay, To appear in Computer Supported Cooperative Work: The Journal of Collaborative Computing, Special Issue on e-Research.
6. Crouchley, R., van Ark, T., Pritchard, J., Grose, D., Kewley, J., Allan, R., Hayes, M. and Morris, L. (2005). Putting Social Science Applications on the Grid. In Proc 1st Int. Conf. on e-Social Science, Manchester.
7. David, P.A. and Spence, M. (2003). Towards institutional infrastructures for e-Science: the scope of the challenge. Research Report No. 2 (Oxford Internet Institute).
8. Dutton, W. (2005). The Internet and Social Transformation: Reconfiguring Access, pp. 375-97 in Dutton, W. H. et al. (eds), Transforming Enterprise. Cambridge, Massachusetts: MIT Press.
9. Dutton, W. D., Jirotka, M. and Schroeder, R. (2006). Ethical, Legal and Institutional Dynamics of the e-Sciences. OeSS Project Draft Working Paper. University of Oxford.
10. Edwards, P., Preece, A., Pignotti, E., Polhill, G. and Gotts, N. (2005). Lessons Learnt from Deployment of a Social Simulation Tool to the Semantic Grid. In Proc 1st Int. Conf. on e-Social Science, Manchester.
11. Fraser, M., Biegel, G., Best, K., Hindmarsh, J., Heath, C., Greenhalgh, C. and Reeves, S. (2005). Distributing Data Sessions: Supporting remote collaboration with video data. In Proc 1st Int. Conf. on e-Social Science, Manchester.
12. French, A., Greenhalgh, C., Crabtree, A., Wright, M., Hampshire, A. and Rodden, T. (2006). Software replay tools for time-based social science data,
13. Guy, M. and Tonkin, E. (2006). Folksonomies: Tidying up Tags? D-Lib Magazine 12(1).
14. Kirschner, P., Buckingham Shum, S. and Carr, C. (2003). Visualising Argumentation: Software Tools for Collaborative and Educational Sense-Making. Springer-Verlag: London.
15. Pignotti, E., Edwards, P., Preece, A, Gotts, M. and Polhill, G. (2005). Semantic Support for Computational Land-Use Modelling. In Proc 5th IEEE International Symposium on Cluster Computing and Grid, Cardiff.
16. Schroeder, R., Caldas, A., Mesch, G. and Dutton, W. (2005). In Proc 1st Int. Conf. on e-Social Science, Manchester.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Towards a Bell-Curve Calculus for e-Science

Lin Yang[1] **Alan Bundy**[1] **Dave Berry**[2] **Conrad Hughes**[2]

[1] University of Edinburgh [2] National e-Science Centre

Abstract

Workflows are playing a crucial role in e-Science systems. In many cases, e-Scientists need to do average case estimates of the performance of workflows. Quality of Service (QoS) properties are used to do the evaluation. We defined the Bell-Curve Calculus (BCC) to describe and calculate the selected QoS properties. The paper presents our motivation of using the BCC and the methodology used during the developing procedure. It also gives the analysis and discussions of the experimental results from the ongoing development.

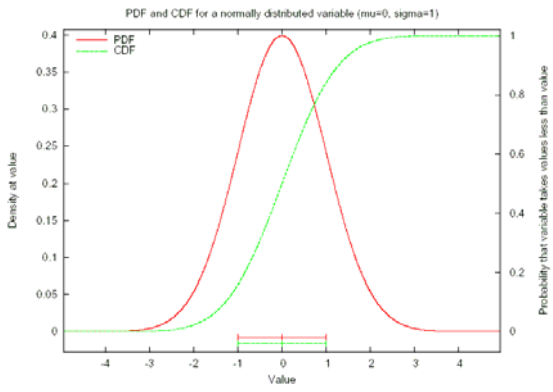
1. Introduction

Grid computing has an almost ten-year history since it was derived, from an analogy to the power Grid, to denote a proposed distributed computing infrastructure for advanced science and engineering collaborations [1]. It is strongly required by consumers, scientists, engineers, enterprises, countries, and even the whole world to share resources, services and knowledge [2]. This sharing is supported and implemented by web services, software systems designed to support interoperable machine-to-machine interaction over a network. These services can be composed in many different ways to form workflows. It is very helpful to measure the performance of the resulting composite services because their quality affects the quality of the Grid directly.

In scientific workflows, experimental data is generated and propagated from one service to another. It would be useful to get rough estimates of various QoS properties, e.g. reliability, accuracy, run time, etc. so that e-Scientists could perform analyses and evaluations of either services or the data produced. We have previously thought about the use of interval arithmetic to calculate error bounds on such estimates. The idea is to extend a numeric value to a number interval, e.g. we use an interval, say [41, 43], to represent the range of error of 42. Extended numeric analysis is used as the way of interval propagation in workflows. The simplest example is for a unary and monotonically increasing function f(x), the extended function f*([a, b]) = [f (a), f (b)]. Using interval arithmetic and propagating error bounds will calculate the biggest accumulative error during workflow executions, so it is a good method for doing a *worst-case* analysis.

However, in more common situations, e-Scientists may want to know the likelihood of each value in the interval. So for *average-case* analysis, we propose to use normal distributions (bell curves) to add the concept of probability to differentiate the likely from the unlikely values of the QoS properties. That is, if we associate a probability density function (pdf) shaped as a bell curve with the estimate, then some values in the interval have a higher probability than others. Figure 1 defines and illustrates the pdf and cumulative density function (cdf) of a standard bell curve.

FIGURE 1
STANDARD BELL CURVE



The graph shows a standard bell curve with parameters – mean value μ=0 and standard deviation σ=1. The red curve is the pdf (probability density function) curve, indicating the probability of each possible value of variable x. It can be generally presented as $p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. The green curve is the cdf (cumulative density function) curve, integrated from its pdf. It gives the probability that a normally distributed variable will output a value ≤ x.

So now the questions are:

(1) Can we use BCC to describe QoS properties and what are the advantages and disadvantages?

(2) How can we define a BCC?

We aim to prove the hypothesis:

The Bell-Curve Calculus is a good estimate of Quality of Service properties over a wide range of values.

2. Why a Bell-Curve Calculus

Although PEPA [3] and some other projects use exponential distribution as their atomic distribution, we still have sufficient reasons to choose bell curve. Initial experimental evidence from DIGS [1] suggests that bell curves are a possible approximation to the probabilistic behaviour of a number of QoS properties used in e-Science workflows, including the reliability of services, considered as their mean time to failure; the accuracy of numerical calculations in workflows; and the run time of services. Moreover, the Central Limit Theorem (CLT) [4] also gives us some theoretical support by concluding that:

"*The distribution of an average tends to be Normal, even when the distribution from which the average is computed, is decidedly non-Normal.*"

Here in the CLT, 'Normal' refers to a Normal Distribution, i.e. a Bell Curve.

Furthermore, from the mathematical description of bell curves, we can see that the biggest advantage of using a bell curve is that its probability density function (pdf) p(x) $= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ has only two parameters: mean value μ and standard deviation σ, where μ decides the location of a bell curve and σ decides the shape of a bell curve. While evaluating the performance of a workflow, we need to gather all the possible data values of the QoS properties we analyse from all the input services. We do calculations and analysis using the information and pass the results through the whole workflow. It will be a big burden if we transfer and calculate all the possible data values one by one. Now using bell curves which have only two parameters, the job becomes more efficient. All we need to do is to store and propagate the two parameters in workflows and a bell curve can be constructed at any time from μ and σ.

Then we will see if we can calculate the QoS properties of a whole workflow from the corresponding properties of its component services, namely if we can define some inference rules to derive the QoS properties of composite services from the correlative properties of their components.

We consider four fundamental methods to combine Grid services (we use services s_1 and s_2 to represent two arbitrary services).

Sequential: s_2 is invoked after s_1's invocation and the input of s_2 is the output of s_1.

Parallel_All: s_1 and s_2 are invoked simultaneously and the outputs are both passed to the next service.

Parallel_First: The output of whichever of s_1 and s_2 first succeeds is passed to the next service.

Conditional: s_1 is invoked first. If it succeeds, its output is the output of the workflow; if it fails, s_2 is invoked and the output of s_2 is the output of the whole workflow.

In terms of the three QoS properties and four combination methods, we have twelve fundamental combination functions (see Table 1). For instance, the combination function of run time in sequential services is the sum of the run times of the component services.

TABLE 1
THE TWELVE FUNDAMENTAL COMBINATION FUNCTIONS

	Seq	**Para_All**	**Para_Fir**	**Cond**
run time	sum	max	min	cond1
accuracy	mult	combine1	varies?	cond2
reliability	mult	combine2	varies?	cond3

The table shows the twelve fundamental combination functions in terms of three QoS properties and the four basic combination methods. Sum, max, min and mult represent respectively taking the sum, maximum, minimum and multiplication of the input bell-curves. Cond1-3 are three different conditional functions and their calculation depends on the succeeding results. The functions of Varies are parallel_first, which means the output of the workflow is the output of the first succeeded service. Combine1-2 are probabilistic merges, which are in the forms of linear-_combinations_of_distribution_1*probability_of_1_occurring+linear_combinations_of_distribution_2*probability_of_2_occurring+...+linear_combinations_of_distribution_N*probability_of_N_occurring. Neither are uniquely defined functions, but depend on different use cases of workflows, which adds the uncertainty to the calculus. But in most workflows, only the basic combinations sum, max, min and mult are needed. What we do for combine1-2 is to combine these basic functions based on different workflow.

Here we convert the formula of bell curve to a function in terms of μ and σ, then get the bell curve function as

[1] DIGS (Dependability Infrastructure for Grid Services, http://digs.sourceforge.net/) is an EPSRC-funded project, to investigate in fault-tolerance system and other quality of service issues in service-oriented architectures

$$bc(\mu, \sigma) = \lambda x. \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}$$

.

Our job is to define different instantiations of the combination functions applying to different QoS properties and different workflow structures.

3. Methodology

Suppose we have two bell curves corresponding to two services. We present them using a bell curve function defined in Section2 as $bc(\mu_1, \sigma_1)$ and $bc(\mu_2, \sigma_2)$. We need to describe μ_0 and σ_0 using μ_1 , μ_2 , σ_1 and σ_2 . That is, μ_0 = $f_\mu(\mu_1, \mu_2, \sigma_1, \sigma_2)$ and σ_0 = $f_\sigma(\mu_1, \mu_2, \sigma_1, \sigma_2)$ The combination function $bc(\mu_0, \sigma_0)$ is defined as $bc(\mu_0, \sigma_0)$ = $F(bc(\mu_1, \sigma_1), bc(\mu_2, \sigma_2))$ = bc($f_\mu(\mu_1, \mu_2, \sigma_1, \sigma_2)$, $f_\sigma(\mu_1, \mu_2, \sigma_1, \sigma_2)$), which is actually a function in terms of four parameters -- μ_1, μ_2, σ_1 and σ_2.

Therefore we have two main tasks:

(1) Can we find a satisfactory instantiation of $F(bc(\mu_1, \sigma_1), bc(\mu_2, \sigma_2))$ for every situation we are investigating?

(2) How good will our approximations be? 'Good' here means accurate and efficient.

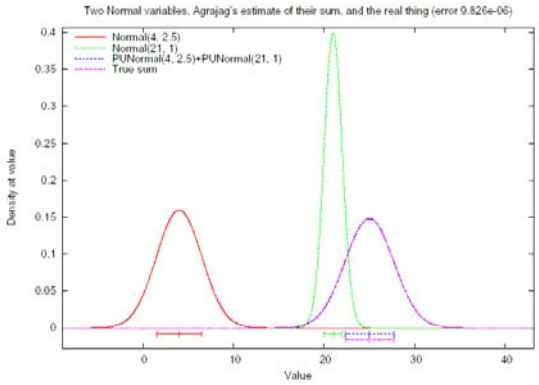
For example, for the property run time in sequential services, we can use $\mu_0 = \mu_1 + \mu_2$ and $\sigma_0 = \sqrt{\sigma_1^2 + \sigma_2^2}$, which has been proved true in mathematics [5].

Our experiments are based on a system called Agrajag [2]. Using Agrajag, we got a satisfactory match (the error is generated by the limited calculation in the approximation method in Agrajag) of the piecewise uniform approximation curve (blue curve) and our estimate curve (mauve curve) (see Figure 2).

Some of our combination functions have been defined by ourselves and tested in Agrajag. For example, for runtime in parallel_all structure, we need to get the maximum of two bell curves. Figure 3 shows the situation of the maximum of two bell curves using the combination method: $\mu_0 = \max(\mu_1, \mu_2)$ and $\sigma_0 = \max(\sigma_1, \sigma_2)$. In this graph, we can see that our estimate achieved a good result – the error is very small. But does it always work like this? When we choose two closer bell curves as the inputs, the error became comparatively large (see Figure 4). This inconsistency decided one aspect worth investigation: through systematic experimentation using Agrajag, we needed to explore in a wide range of data to find various error status in different input situations.

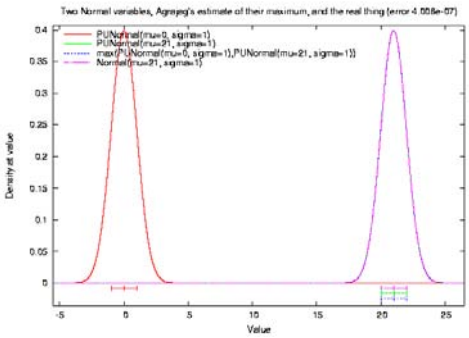
[2] Agrajag (http://digs.sourceforge.net/agrajag.htm) is a framework written in Perl and C, developed by Conrad Hughes, to implement some operations and measurements on some basic models of stochastic distributions.

FIGURE 2
THE SUM OF TWO BELL CURVES
AND ITS APPROXIMATION



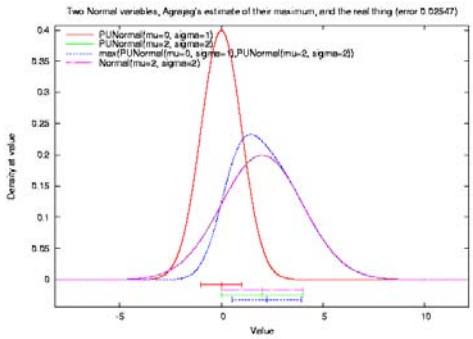
The graph shows the sum of two bell curves (red curve and green curve). It can be used to model the run time of sequential combinations. Here we use an exact mathematically proved method: $\mu_0 = \mu_1 + \mu_2$ and $\sigma_0 = \sqrt{\sigma_1^2 + \sigma_2^2}$ to estimate the piecewise uniform curve (blue curve) produced by Agrajag. The mauve curve is our approximation curve, which almost coincides with the blue curve. We can see there is still a tiny error shown at the title of the graph. It is caused by the approximation using piecewise uniform functions. In the ideal situation (the resolution values which divide a curve to locally constant and connected segments $\rightarrow+\infty$), the error is zero.

FIGURE 3
THE MAXIMUM OF TWO BELL CURVES
AND ITS APPROXIMATION (1)



This graph shows an ideal situation of getting the maximum of two bell curves. The red curve and the green curve are the two inputs. In this case, using the method $\mu_0 = \max(\mu_1, \mu_2)$ and $\sigma_0 = \max(\sigma_1, \sigma_2)$, the green curve is the piecewise uniform form of our approximation, the mauve curve. The blue curve is Agrajag estimate. Since the green curve, the blue curve and the mauve curve almost coincide with each other, they are hardly distinguished in the figure.

FIGURE 4
THE MAXIMUM OF TWO BELL CURVES
AND ITS APPROXIMATION (2)



In this graph, we use the same combination method as that in Figure 3, but taking two much closer bell curves as the inputs. This time, there is a distinct difference between Agrajag's estimate (the blue curve) and our approximation (the mauve curve). We can see that in this case, the mauve curve and the green curve are the same curve, so they coincide with each other.

Another investigation aspect is to define and compare all sorts of combination methods to get, say, the maximum of bell curves. For example, through testing in Agrajag, we discovered that in most common situations, to get the maximum of two bell curves, the effect of approximating the output curve using $\sigma_0 = \max(\sigma_1, \sigma_2)$ is better than that using $\sigma_0 = \sqrt{\sigma_1^2 + \sigma_2^2}$ or that using $\sigma_0 = 1/(1/\sigma_1 + 1/\sigma_2)$. Our main goals are to make comparisons of all sorts of approximation methods and find the best one across many different situations.

To achieve a better outcome, we need some methods to define the precision of the approximation methods we use and then refine the experimental results. The explicit way to get to precision is to calculate the average error values, which allows us to have a general idea about how accurate our approximation is and make a comparison between different approximation methods easily. However, it cannot indicate how we could improve our method to get a better result. In Agrajag, there is a functionality to derive the parameters of the piecewise uniform approximation to the combination functions. So we call these parameters the perfect parameters and use them as a standard. Then we transform the job of finding the most suitable parameters of the combination functions to matching the perfect parameters. We will elaborate it using an example in Section 4.

All the above description to our methodology raises the question: since Agrajag can perform piecewise uniform approximation of bell-curve combinations, why do we still need a BCC? Why don't we just use Agrajag to produce a bell-curve approximation to a workflow using the data from its component services? The answer is efficiency. Agrajag's calculations do well in small workflow calculations, but the more common scenario is that workflows sometimes are composed of thousands of services. To take all the inputs and get an approximation requires huge calculation capacity, which will make Agrajag's runtime unacceptably long. While using the BCC, we just need to do calculations among the parameters, which will make the calculation procedure more efficient.

4. Experimental Result and Analysis

In this section, we will give some experimental results and analysis according to the methodology we have described in Section 2 and Section 3. Since the combination function of sum of two bell curves is exact, we make our first attempt on the method of the getting maximum of two bell curves, which does not have a known simple mathematical combination.

To get more intuitive results, we used Gnuplot[3] to draw 3D graphs. Without loss of generality, we fixed one of the input bell curves to the standard bell curve (μ_2 =0 and σ_2 =1). Then the three dimensions were set as μ_1, σ_1 and the difference between the piecewise uniform estimation and our approximation using our combination methods. To ensure that common situations are considered, we generated $bc(\mu_1, \sigma_1)$ from a range of logarithmic-scaled integers, e.g., $2^{-10} \le \mu_1 \le 2^{10}$ and $2^{-10} \le \sigma_1 \le 2^{10}$.

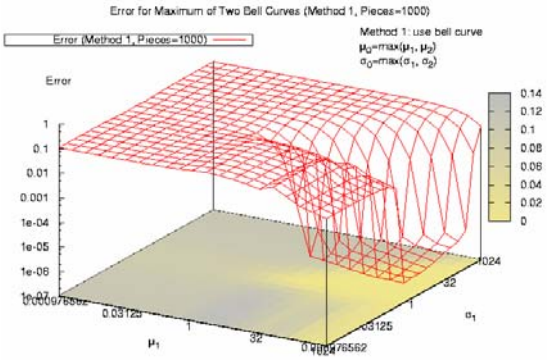
Figure 5 shows the experimental results using a combination method (Method 1): $\mu_0 = \max(\mu_1, \mu_2)$ and $\sigma_0 = \max(\sigma_1, \sigma_2)$. From this graph we can see how the value of the error changes. Especially in the area of $7 \le \mu_1 \le 9$ and $1 \le \sigma_1 \le 1.6$, the errors are near 1e-06, which is a quite satisfactory approximation.

Does the method shown in Figure 5 achieve the best result? We tested another method (Method 2): $\mu_0 = \max(\mu_1, \mu_2)$ and $\sigma_0 = \sigma_1$ or σ_2 (with bigger μ) (see Figure 6). In Figure 6, we can see that the area of tiny errors is extended, compared to Figure 5. In most areas, the two surfaces coincide with each other, which is

[3] Gnuplot is a portable command-line driven interactive data and function plotting utility for many operating systems. It can plot either 2D or 3D graphs.

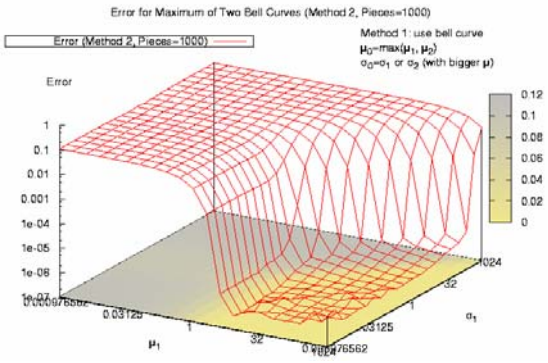
always true when $\sigma_1 \geq 1$ because $\sigma_2 \equiv 1$ and both methods will take $\sigma_0 = \sigma_1$. Whereas in the area $\mu_1 \geq 4$ and $\sigma_1 < 1$, the green surface (Method 2) is much lower than the red one (Method 1). But the two methods are still the best two among all the methods we tried. Table 2 shows all the combination methods we had tried to get the maximum of two bell curves and their average errors. For all the methods we used $\mu_0 = \max(\mu_1, \mu_2)$.

FIGURE 5
THE MAX OF TWO BELL CURVES
AND ITS APPROXIMATION – METHOD 1



The graph shows the error distribution for $2^{-10} \leq \mu_1 \leq 2^{10}$ and $2^{-10} \leq \sigma_1 \leq 2^{10}$. The X-axis is the value of μ_1, the Y-axis is the value of σ_1 and the Z-axis is the value of error between the piecewise uniform estimate (the resolution values is 1000 for this case) and the BCC estimate using the method $\mu_0 = \max(\mu_1, \mu_2)$ and $\sigma_0 = \max(\sigma_1, \sigma_2)$. A lower value for the Z-axis shows a better fit. The values of x, y and z are discrete, but we set it drawn with the parameter 'by steps', which allows Gnuplot to connect every two points and give us a clearer figure. The colourful platform map at the bottom of the coordinates indicates the various values of error. From the label on the right, we can see that from grey to yellow, the value of error decreases.

FIGURE 6
THE MAX OF TWO BELL CURVES
AND ITS APPROXIMATION – METHOD 2



The graph shows the error distribution using the method $\mu_0 = \max(\mu_1, \mu_2)$ and $\sigma_0 = \sigma_1$ or σ_2 (with bigger μ). Please note that the yellow areas in the platform map do not imply that all the values of the error are zero, but rather the errors are too small to distinguish from zero.

When we observe the above three figures, we can see that the errors produced by both methods stay stable at a comparatively high value in some areas. For instance, in Figure 6, there are some areas with correspondingly high error values and a sharp descent on error values at $\mu_1 \approx 4$. Why is there a distinct difference among the values? We did an experiment using method 2 to get the answer.

We set the numbers of pieces of piecewise uniform functions as 10, 100 and 1000 and got the three piecewise uniform estimates of the maximum of two bell curves. Then we used method 2 to derive our approximation of the maximum and obtained three error distributions. We drew the three distributions in one graph (Figure 7). We can see that the high-error areas of the three distributions coincided with each

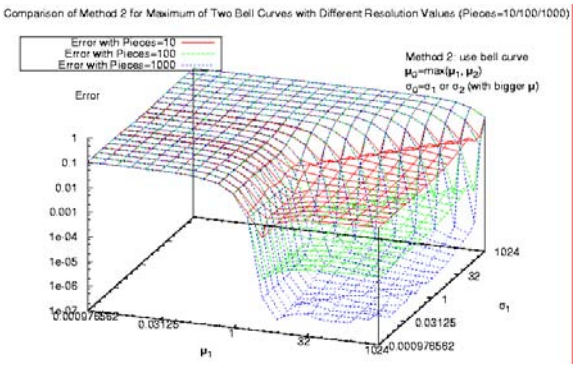
TABLE 2
THE COMBINATION METHODS OF GETTING MAXIMUM OF TWO BELL CURVES

Method	Average error	Method	Average error
$\sigma_0 = \max(\sigma_1, \sigma_2)$	0.0563955	$\sigma_0 = \sigma_1$ or σ_2 (with bigger μ)	0.0550807
$\sigma_0 = \sqrt{\sigma_1^2 + \sigma_2^2}$	0.0782595	$\sigma_0 = 1/(1/\sigma_1 + 1/\sigma_2)$	0.0799015
$\sigma_0 = \sigma_1 + \sigma_2$	0.0862074	$\sigma_0 = 1/\sigma_1 + 1/\sigma_2$	0.1160483
$\sigma_0 = 0.8 \times \sigma_1 + 0.2 \times \sigma_2$ (with bigger μ)	0.0564251	$\sigma_0 = \sqrt{(\sigma_1^2 + \sigma_2^2)/2}$	0.0676253
$\sigma_0 = \sigma_1 \times \sigma_2$	0.0550807	$\sigma_0 = \sqrt{\sigma_1 \times \sigma_2}$	0.0648305
$\sigma_0 = (\sigma_1 + \sigma_2)/2$	0.0650388	$\sigma_0 = \lvert \sigma_1 - \sigma_2 \rvert \times \sigma_1/\sigma_2 + \lvert \sigma_1 - \sigma_2 \rvert \times \sigma_2/\sigma_1$	0.1194134

The table shows the situation when we use different combination methods to get the maximum of two bell curves at the same resolution value. We set $\mu_0 = \max(\mu_1, \mu_2)$ in all the methods. μ_1 And σ_1 both take values from 2^{-5} to 2^5. Since we use the standard bell curve as one input and $\mu_1 > 0$, $\sigma_0 = \sigma_1$ or σ_2 (with bigger μ) and $\sigma_0 = \sigma_1 \times \sigma_2$ are the same method in this case. Despite this, the first two combination methods are the best two methods we got. The combination methods we choose are rough hypotheses based on Figure 4. We estimate the output parameters according to the location and shape of the Agrajag approximation curve. We calculated the average error of each method to compare how good these combination methods are.

other. But since the three distributions used different resolution values, which means that the precisions of the three calculations are different and there should be a minimum difference on the error values with different numbers of pieces. While in some areas, the value is almost unchanged, which means the method we used did not get a correct result in these areas. We tested all the methods we used in our experiment and could not find a completely satisfactory method.

FIGURE 7
THE COMPARISON OF COMBINATION METHOD 2
WITH DIFFERENT RESOLUTION VALUES



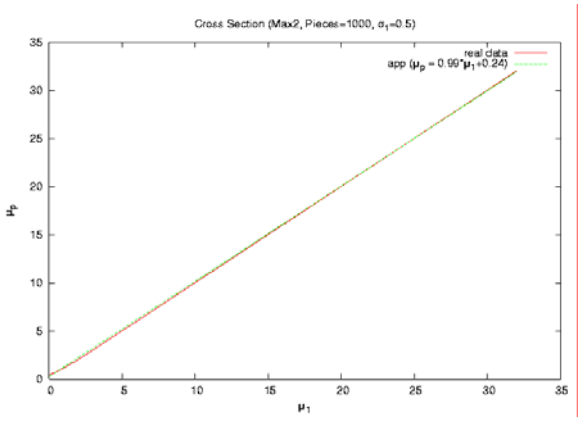
The graph provides us the comparison of error distributions when we separately take the number of a pieces in piecewise uniform estimate as 10, 100 and 1000. Here the method is $\mu_0 = \max(\mu_1, \mu_2)$ and σ_0 = σ_1 or σ_2 (with bigger μ).

In this case, we could use our perfect parameters method stated in Section 3 to facilitate fine adjustments on our approximation functions.

We derive the perfect parameters μ_p and σ_p in Agrajag and try to approximate them in terms of μ_1 , μ_2 , σ_1 and σ_2 . Then the combination function we aim for turns to $bc(\mu_p, \sigma_p)$ = bc($f_\mu(\mu_1, \mu_2, \sigma_1, \sigma_2)$, $f_\sigma(\mu_1, \mu_2, \sigma_1, \sigma_2)$) $F(bc(\mu_1, \sigma_1), bc(\mu_2, \sigma_2))$.

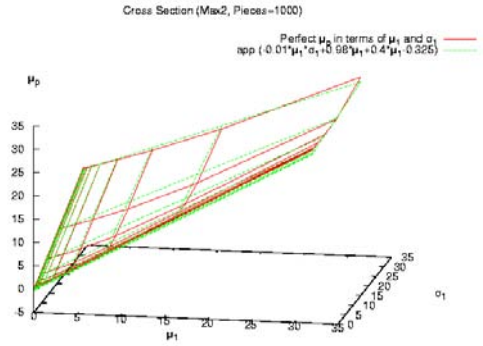
We fix one parameter, such as σ_1, then use some function to describe the relation between μ_p and μ_1. To simplify the problem, we use linear function here, namely $\mu_p = a \times \mu_1 + b$. Then we get sets of a and b corresponding to different σ_1 value. Figure 8 gave us the linear approximation when σ_1 = 0.5. Then we could have the linear function of σ_1 in terms of a and b. So we finally get the function of μ_p in terms of μ_1 and σ_1 (see Figure 9).

FIGURE 8
THE LINEAR APPROXIMATION OF μ_p



The red line is the real data of μ_p and the green line is our approximation using the linear function $\mu_p = 0.99 \times \mu_1 + 0.24$. In this case, σ_1 is fixed to 0.5. In this figure, the linear approximation achieved a good result, especially in the upper part of the curve.

FIGURE 9
THE APPROXIMATION OF μ_p IN TERMS OF μ_1 AND σ_1

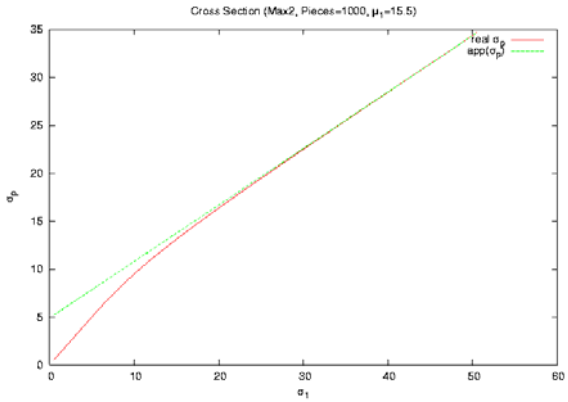


In this graph, the red surface is the real perfect μ_p surface and the green surface is our approximation of μ_p in terms of μ_1 and σ_1. We can see in this figure, the two surfaces do not coincide in all the areas, which means that we still need to do some adjustment during the procedure of the linear approximation. We could change parameters of the linear functions or use non-linear functions to do the approximation.

In the above example, only using linear approximation seems to get a good result. However, in most cases, linear functions alone cannot achieve a satisfactory result. For example, this time we fix μ_1 to 15.5, then we use a linear function of σ_1 to approximate σ_p (see Figure 10). There is a big gap between the curve and the approximation. After drawing the difference curve, based on the curve shape, we used different compensation to approximate it. The choices of exponential or other functions were made by our experience and the

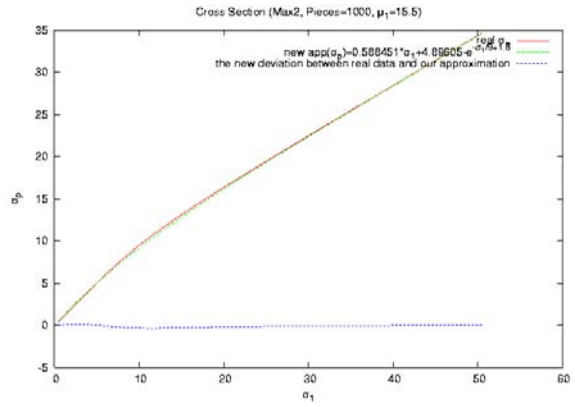
parameters of those functions were chosen by fitting the curves gradually. Figure 11 shows the result after adding the exponential compensation. The error between the real perfect values and our approximation is reduced appreciably. We could repeat the above procedure until the result reaches the acceptable range.

FIGURE 10
THE LINEAR APPROXIMATION TO σ_p
IN TERMS OF σ_1 WHEN μ_1 IS FIXED



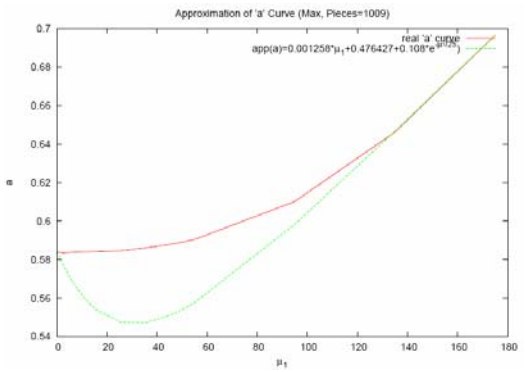
In this graph, we fixed μ_1 = 15.5 and used a linear function of σ_1 to approximate σ_p. The green line is our approximation, whereas the red curve is the real σ_p data. We can see the lower part of the curve dropped remarkably. So only using linear approximation does not produce a satisfactory result.

FIGURE 11
THE APPROXIMATION OF σ_p IN TERMS OF σ_1
AFTER ADDING EXPONENTIAL COMPENSATION



In this figure, the red curve is the real σ_p data. The green curve is our approximation using linear function plus some exponential compensation. The blue curve is the error between the real data and our approximation, which is quite flat to the naked eye. Compared to Figure 10, we can see that our improved approximation method achieved a much better result. If we require more precise result, we could add more compensation to the method.

FIGURE 12
THE APPROXIMATION OF 'A' IN TERMS OF μ_1



In this graph, the red surface is the real perfect 'a' surface and the green surface is our approximation of 'a' in terms of μ_1. There is a gap between the two curves in the area when μ_1 is relatively small. We could still use some compensation function to erase it. But in this case, the absolute value of error is acceptable. So we chose to keep the form of the approximation function simpler.

We applied the above procedure to different μ_1 values and found that σ_p could always be approximated by the function f(σ_1) in the form of f(σ_1) = a* σ_1 +b-exp(- σ_1 /d) and the parameters a, b, c and d vary regularly along the values of σ_1. So we present the parameters a, b, c and d in term of μ_1 separately. Figure 12 shows the situation when we approximate the parameter a using a function of μ_1.

After deriving all the four functions of approximating the parameters a, b, c and d in term of μ_1, we substitute them into the perfect function σ_p = f(σ_1). Then we have the final approximation function of σ_p in terms of both μ_1 and σ_1. We repeat the procedure to get the approximation function of μ_p, which is more precise than the result in Figure 9.

5. Use Case

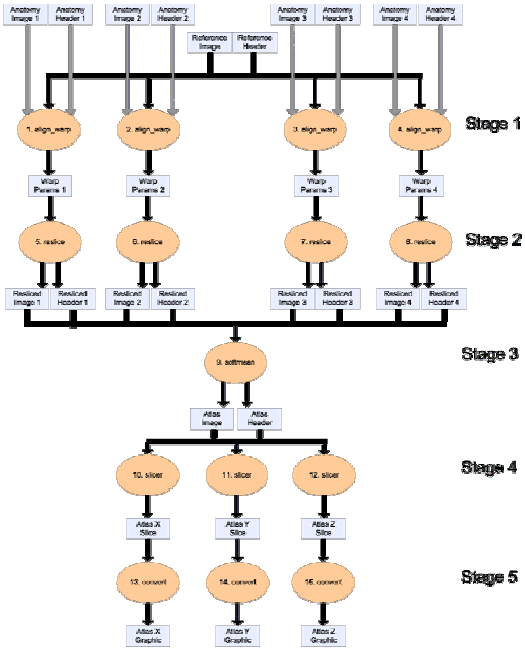
After deriving the fundamental combination functions for runtime, we apply the BCC to some use case (Figure 13), which can be abstracted using our combination functions (Figure 14).

We use Agrajag results as the gold standard values against which to evaluate our BCC. Figure 16 shows the difference between the standard values and our approximation values. Through calculating the runtime of the whole workflow, we could:

(1) Do evaluations on the performance of the workflow;

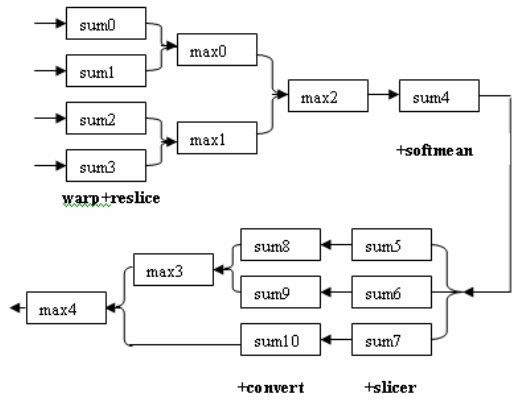
(2) Provide measurement values for e-Scientists to make decisions on whether to use the workflow or not.

FIGURE 13
EVALUATION USE CASE



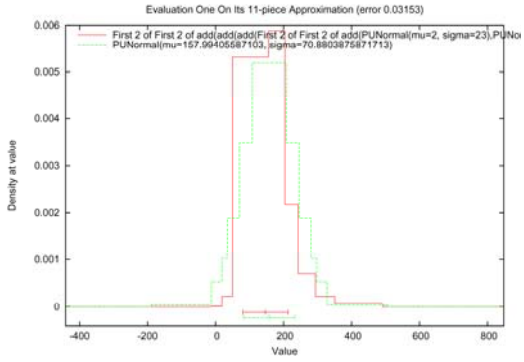
This is an example workflow for creating population-based "brain atlases", comprised of *procedures*, shown as orange ovals, and *data items* flowing between them, shown as rectangles. More details of the workflow can be found in http://twiki.ipaw.info/bin/view/Challenge/FirstProvenanceChallenge.

FIGURE 14
ABSTRACTED WORKFLOW USE CASE



This graph shows the abstract structure of the workflow using our fundamental combination functions. In this case, the workflow contains only the structures of 'Seq' and 'Para_All', so only 'sum' and 'max' are concerned.

FIGURE 15
APPROXIMATION OF RUNTIME



The graph presents the runtime of the workflow in Figure 13. In this graph, the blue curve shows the Agrajag values, namely, the standard values; while the green curve is our approximation values. Due to the low resolution, the difference is acceptable. If we raise the resolution, the advantage of the BCC will be obvious. It will take much less time to achieve the values.

6. Future Work

In Section 3~5, we gave the methodology and some experimental results of the BCC, and also applied it to certain use case. The next step is to complete the BCC by finishing the rest of the fundamental combination functions. Then we will find real data to do more evaluation. After that, we will consider extend the BCC by importing Log-Normal or other distributions to describe the values more precisely. We may also embed the extended BCC to some frameworks to enhance their functionalities of prediction and evaluation.

References

[1] Foster,I., Kesselman,C., The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, 1999
[2] Smarr, L., Chapter 1. Grids in Context, in The Grid 2, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003
[3] Hillston, J. A Compositional Approach to Performance Modelling 1995
[4]http://www.statisticalengineering.com/central_limit_theorem.htm
[5]http://mathworld.wolfram.com/NormalSumDistribution.html

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Dynamic Discovery of Composable Type Adapters for Practical Web Services Workflow

Martin Szomszor, **Terry R. Payne** and Luc Moreau
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
{mns03r, trp, L.Moreau}@ecs.soton.ac.uk

Abstract

As the Web Services and Grid community adopt Semantic Web technology, we observe a shift towards higher-level workflow composition and service discovery practices. While this provides excellent functionality to non-expert users, more sophisticated middleware is required to hide the details of service integration. By investigating a bioinformatics use case, we observe the need for Type Adaptor *components to be inserted into workflows to harmonise syntactically incompatible interfaces. In this paper, we propose a generic Type Adaptor description that can be used in conjunction with existing service registries to facilitate automatic syntactic mediation. We demonstrate our implementation before evaluating both the translation approach we employ, and the relative cost of using a registry for Type Adaptor discovery.*

1. Introduction

Workflow technology has been adopted by e-Science Grid applications to encode scientific processes, allowing users to perform *in silico* science [7]. The MYGRID (www.mygrid.org.uk) project is an example of such a system, supporting bioinformaticians in the construction, execution and sharing of workflows through the Taverna (http://tavera.sf.net) graphical workbench. Recent advances in MYGRID have focused on supporting users in the discovery and composition of services by adding rich service annotations. FETA [12] has incorporated Semantic Web [3] technology into service descriptions using ontologies to capture the semantics of Web Service behaviour - essentially supplying users with conceptual definitions of what the service does using domain specific terminology. This has proven to be a valuable commodity in a system potentially making use of thousands of services where searching over service descriptions alone is a cumbersome and tedious task.

With the introduction of semantically annotated Web Services, workflow composition in MYGRID has shifted to a higher-level design process: bioinformaticians can choose to include services in a workflow to achieve particular goals based on conceptual service definitions. While this makes workflow design more accessible to untrained users, it does lead to more complex architectural requirements. The situation often arises where users wish to connect two services together that are *conceptually compatible* but have different *syntactic interfaces*. To harmonise any data incompatibilities in a workflow, additional processing is required, often taking the form of translation script, bespoke application, or Web Service. Within MYGRID, these *Type Adaptor* components must be discovered manually and inserted into the workflow by hand, imposing additional effort on the bioinformatician. Consequently, they are distracted from the scientific process at hand, spending additional time understanding why an incompatibility has been encountered and how it can be harmonised.

In this paper, we present a generic approach for describing Type Adaptors, which separates abstract functionality (i.e. the data types converted) from implementation (translation script, executable code, Web Services, etc). By using such a Type Adaptor description with a service registry, we show that it is possible to advertise Type Adaptors and discover them at run-time, aiding the user by automatically including the necessary conversion components. By combining the implementation of a composable translation language with the Grimoires service registry (www.grimoires.org), our proposed architecture is able to automate the discovery and execution of Type Adaptors in Web Service workflows. We evaluate our implementation to show that the translation approach scales well, offers composability with little cost, and results in a relatively low overhead when used with the Grimoires registry. Our con-

tributions include:

1. An approach for Type Adaptor interface description in WSDL using Grimoires for advertising and discovery;
2. A language to describe Type Adaptors and an engine to process them and perform type conversion;
3. A complete system implementing the approach with empirical evaluation through benchmarking.

This paper is organised as follows: Section 2 introduces the need for Type Adaptors and presents a motivating example within a bioinformatics scenario. In Section 3, the use of WSDL for Type Adaptor description and discovery is shown before we present our implementation in Section 4. Evaluation of our implementation is given in Section 5, followed by the examination of related work in Section 6. Finally, we conclude and present further work in Section 7.

2. Motivation and Use Case

A typical task within bioinformatics involves retrieving sequence data from a database and passing it to an alignment tool to check for similarities with other known sequences. Within MYGRID, this interaction is modelled as a simple workflow, with each stage in the task being fulfilled by a Web Service, illustrated in Figure 1. Many Web Services are available for retrieving sequence data; the ones used here are DDBJ-XML (`http://xml.ddbj.nig.ac.jp/`) and XEMBL (`http://www.ebi.ac.uk/xembl/`). To obtain a sequence data record, an accession number (unique id) is passed as input to the service, which returns an XML document. This document, returned from either service, essentially contains the same information, namely the sequence data as a string (e.g. `atgagtga...`), references to publications, and features of the sequence (such as the protein translation). However, the way this information is represented differs - XEMBL returns an INSD[1] formatted record whereas DDBJ-XML returns a document using their own custom format. The next stage in the Workflow is to pass the sequence data to an alignment service, such as the BLAST service at NCBI[2]. This service can consume a string of FASTA[3] formatted sequence data.

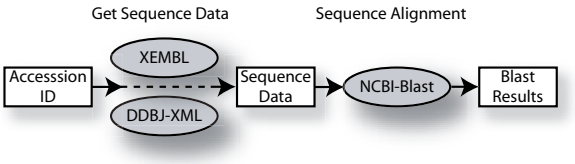


Figure 1. A simple bioinformatics task: get sequence data from a database and perform a sequence alignment on it.

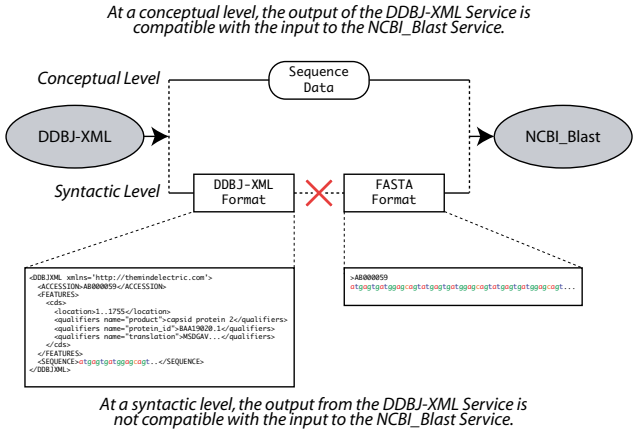


Figure 2. The output from the DDBJ-XML Service is not compatible for input to the NCBI-Blast Serivce.

Intuitively, a Bioinformatician will view the two sequence retrieval tasks as the same type of operation, expecting both to be compatible with the NCBI-Blast service. The semantic annotations attached through FETA affirm this as the output types are assigned the same conceptual type, namely a `Sequence Data Record` concept. However, when plugging the two services together, we see that the output from either sequence data retrieval service is not directly compatible for input to the NCBI-Blast service (Figure 2). To harmonise the workflow, some intermediate processing is required on the data produced from the first service to make it compatible for input to the second service. We define this translation step as *syntactic mediation*, an additional workflow stage that is carried out by a particular class of programs we define as *Type Adaptors*.

3. Generic Type Adaptor Description

To augment the manual selection of programs or services to harmonise data incompatibles in Web Service workflows, we propose a solution that utilises a registry of Type Adaptors, each of them described by WSDL, to support the automated discovery of harmonisation components. In this Section, we characterise a generic approach for the description, sharing and discovery of Type Adaptors and how they can be used to perform syntactic mediation in workflows.

There are many applications and tools that support the translation of data between different formats. XSLT [6] enables the specification of data translation in a script for-

[1] http://www.ebi.ac.uk/embl/Documentation/DTD/INSDSeq_v1.3.dtd.txt

[2] http://www.ncbi.nlm.nih.gov/BLAST/

[3] http://www.ebi.ac.uk/help/formats_frame.html

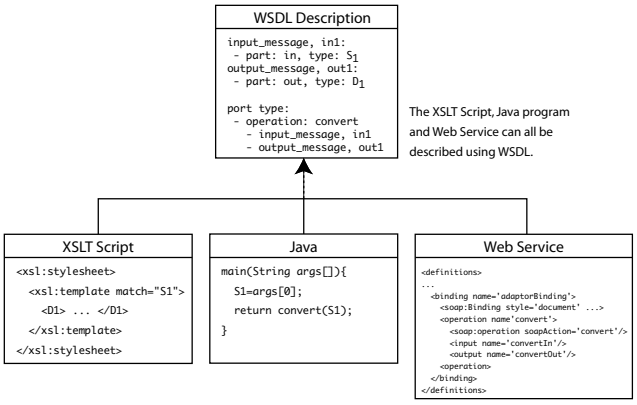


Figure 3. Using WSDL to describe different Type Adaptors

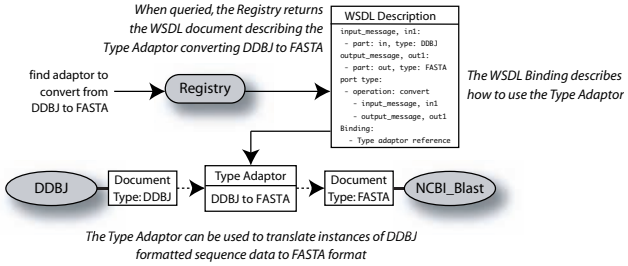


Figure 4. The use of a registry to discover Type Adaptors

mat using pattern matching and template statements. Such a script can be consumed by an XSLT engine to drive the translation of data to a different representation. Other forms of Type Adaptors are not so transparent; a black box approach is used frequently in MYGRID where users create custom translation programs using languages such as JAVA and PERL. In other cases, a Type Adaptor may take the form of a distinct mediator Web Service, described by WSDL and executed using SOAP over HTTP. Any of these Type Adaptors can be viewed as a component that converts data from a source type to a destination type. To describe the capabilities of all Type Adapters, irrespective of implementation, we separate concrete implementation details from the abstract definition. Under this assumption, all Type Adaptors can be described using WSDL [5].

WSDL is a declarative language used to specify service capabilities and how to access them through the definition of service end-points. The operations implemented by the service are defined in terms of the messages consumed and produced, the structure of which is specified by XML Schema. The service, operations and messages are described at an abstract level and bound to a concrete execution model via the service binding. The service binding describes the type of protocol used to invoke the service and the requested datatype encoding. Because of this two-tier model, many different Web Service implementations may be viewed through a common interface. By applying the same principle to data harmonisation components, we can use WSDL to describe the capabilites of any Type Adaptor. Using this approach allows different implementations of the same Type Adaptor to be described with the same abstract definition (i.e. in terms of the input and output XML schema types) and different bindings. This is illustrated in Figure 3, where three Type Adaptors are shown: an XSLT script, a JAVA program and a SOAP Web Service, all providing the same functionality - to convert data of type S_1 to D_1. Although other Web technologies, such as RDF [10], would be adequate for describing Type Adaptor behaviour in this way, WSDL is standardised and widely used in other Web Service technologies (e.g. the workflow language WSFL [11] and the choreography language WS-CDL [9]), and would therefore facilitate technology reuse in future work.

With a uniform method for the description of Type Adaptors in the form of WSDL, we can utilise existing registry technologies to support sharing and discovery - this feature is described in more detail in Section 4 where we present our implementation. Figure 4 shows a high level view of how a registry containing WSDL definitions of Type Adaptors can be used in a Web Service workflow to perform syntactic mediation. The output from the DDBJ Service, of XML type DDBJ, is used as input to the NCBI-Blast Service, which consumes type FASTA. The binding section of the WSDL definition describes how to execute the translator, for example, by providing the location of an XSLT script or the JAVA method details.

4. Architecture

In this Section, we describe an architecture that utilises the Type Adaptor description outlined above. After presenting the motivation for an intermediary representation, we briefly describe our mapping language, FXML-M, that is used to specify Type Adaptor behaviour, present our Type Adaptor registration and discovery implementation, and show it working against our use case scenario.

4.1. Scalability and Reuse

As the amount of Web Services is increasing, currently over 1000 in MYGRID [12, 8], scalability and reuse is an important issue. In the simplest case, we assume a Type Adaptor exists to convert data directly between every compatible representation. For n compatible data formats, $O(n^2)$ Type Adaptors are required to achieve maximum interoperability. Therefore, as more Web Services are introduced, the number of compatible interfaces increases and a quadratic expansion for the number of Type Adaptors will occur. Also,

when introducing a new representation for information that is already present in other formats, Type Adaptors must be created to transform the new representation to all other formats.

By introducing an intermediate representation, to which all data formats are converted, the problem of scalability can be diminished. For n compatible interfaces, $O(n)$ Type Adaptors are required, resulting in a linear complexity as more services are added. When introducing new formats, only one Type Adaptor is required to convert the new data format to and from the intermediate representation. Therefore, our Architecture is based around the use of an intermediary representation. When considering the mechanisms necessary to support users in the use and agreement of a common representation, we see existing work already tackles a similar problem within MYGRID. FETA uses descriptions that supply users with conceptual definitions of what the service does using domain specific terminology. Part of this solution involved the construction of a large ontology for the bioinformatics domain covering the types of data shown in our use case. Therefore, we extend these existing OWL ontologies to capture the semantics and structure of the data representation.

In terms of sharing and reuse, it is common for Web Services to supply many operations that operate over the same, or subsets of the same data. Therefore, the transformation for a given source type may come in the form of a Type Adaptor designed to cater for multiple types. Hence, the generation of a WSDL description for a given Type Adaptor should capture *all* the transformation capabilities of the adaptor. This can be achieved by placing multiple operation definitions in the WSDL, one for each of the possible type conversions.

4.2. Bindings

To describe the relationship between XML schema components and OWL concepts / properties, we devised a mapping language, FXML-M (Formalised XML mapping), described in previous work [18]. FXML-M is a composable mapping language in which mapping statements translate XML schema components from a source schema to a destination schema. FXML-T (Formalised XML Translator) is an interpreter for FXML-M which consumes an *M-Binding* (collection of mappings) and the source XML document and produces a destination document. Broadly, an M-Binding B contains a sequence of mappings $m_1, m_2, \ldots, m_n$. In the style of XML schema, M-Bindings may also import other M-Bindings (e.g. $B_1 = \{m_1, m_2\} \cup B_2$) to support composition - an important feature when services offer operations over multiple XML schemas. M-Bindings themselves are XML documents which be viewed as a specialised mini workflow for type conversion.

4.3. Binding Publication and Discovery

Since our M-Binding language is used to specify XML to XML document conversion, and our intermediary language is an OWL ontology, conversion between XML instances and OWL concept instances is specified in terms of a canonical XML representation for OWL individuals. While the use of XML to represent OWL concepts is common, XML Schemas to describe these instances are not available. Hence, we automatically generate XML Schemas (OWL instance schema) to describe valid concept instances for a given ontology.

With an intermediary schema in place (in the form of an OWL instance schema), the appropriate M-Bindings to translate data to and from XML format, and FXML-T (the FXML-M translator), Web Service harmonisation is supported. To fully automate the process, i.e. discover the appropriate M-Bindings (themselves individual Type Adaptors) based on translation requirements, we require a registry to advertise M-Bindings. Since Type Adaptors can be described using WSDL, we use the Grimoires service registry (www.grimoires.org) to record, advertise and supply adaptors. Grimoires is an extended UDDI [1] registry that enables the publishing and querying of Service interfaces while also supporting semantic annotations of service descriptions [14]. Figure 5 illustrates how Bindings are registered with Grimoires using our *Binding Publisher Service*.

The *Binding Publisher Service* consumes an M-Binding, along with the source and destination XML schemas, and produces a WSDL document that describes the translation capability of the M-Binding. This generated WSDL in then publicly hosted and registered with Grimoires using the `save_service` operation. Afterwards, an M-Binding WSDL may be discovered using the standard Grimoires query interface (the `findService` operation) based on the input and output schema types desired.

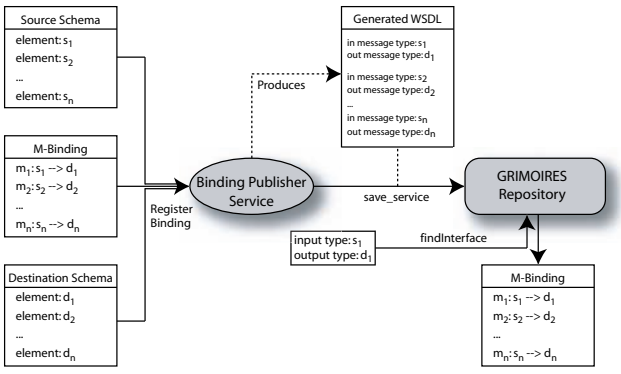


Figure 5. Registration and discovery of Binding using the Grimoires repository

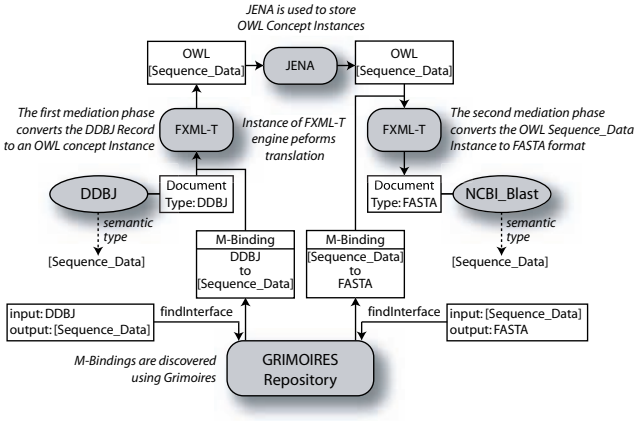


Figure 6. Automated syntatic mediation for our use case using Grimoires to store Binding descriptions and FXML-T for document translation

4.4. Automated Mediation

To achieve syntactic mediation, two M-Bindings are necessary: a *realisation M-Binding* (to convert XML to the intermediate OWL representation), and a *serialisation M-Binding* (to convert OWL to XML). Figure 6 is an expansion of Figure 4 showing the use of Grimoires and FXML-T to automate syntactic mediation for our use case using and intermediate OWL representation. When registering the DDBJ and NCBI-Blast services with FETA, semantic annotation are used to specify the input and output *semantic types* using a reference to a concept in the Bioinformatics ontology - in this case the `Sequence_Data` concept. At workflow composition time, the Bioinformatician may wish to feed the output from the DDBJ service into the NCBI-Blast service because they are deemed semantically compatible (i.e. they share the same semantic type). While this workflow can be specified, it is not executable because of the difference in data formats assumed by each service provider - a stage of syntactic mediation is required. Figure 6 shows the mediation phase, using Grimoires to discover M-Bindings, FXML-T to perform the translation from DDBJ format to FASTA format with JENA used to hold the intermediate representation (in the form of an OWL concept instance).

5. Evaluation

To evaluate this approach, we have examined the performance of our Translation Engine (FXML-T) and the use of Grimoires as a repository for advertising M-Binding documents. The aims are threefold: (i) to test the scalability of our mapping language approach; (ii) to establish FXML-T as scalable translation engine by examining the performance costs against increasing document sizes, increasing schema sizes, and increasingly complex M-Binding composition; (iii) to confirm the use of Grimoires for M-Binding discovery is not a significant overhead in the context of workflow execution. The following sub-Sections describe each of the tests with hypothesis given in *italics*. All tests were carried out using a 2.6 Ghz Pentium4 PC with 1GB RAM running Linux (kernel 2.6.15-20-386) using unix `time` to record program execution times. FXML-T is implemented in SCHEME and run using the Guile Scheme Interpreter (v1.6). Tests in Section 5.1 are run 10 times with a mean value plotted.

5.1. Translation Engine Scalability

Expanding document and schema size will increase the translation cost linearly or better.
We test the scalability of FXML-T in two ways: by increasing input document size (while maintaining uniform input XML schema size), and by increasing both input schema size and input document size. We test FXML-T against the following XML translation tools:

- XSLT: Using Perl and the XML::XSLT module[4].
- XSLT: Using JAVA (1.5.0) and Xalan[5](v2.7.0).
- XSLT: Using Python (v2.4) and the 4Suite Module[6](v0.4).
- SXML: A SCHEME implemention for XML parsing and conversion (v3.0).

Since FXML-T is implemented using an interpreted language, and Perl is also interpreted, we would expect them to peform slowly in comparison to JAVA and Python XSLT which are compiled[7]. Figure 7 shows the time taken to transform a source document to a structurally identical destination document for increasing document sizes.

The maximum document size tested is 1.2 MB, twice that of the Blast results obtained in our use-case. From Figure 7 we see that FXML-T has a linear expansion in transformation time against increasing document size. Both Python and JAVA implementations also scale linearly with better peformance than FXML-T due to JAVA and Python using compiled code. Perl exhibits the worst performance in terms of time taken, but a linear expansion is still observed.

Our second performance test examines the translation cost with respect to increasing XML schema size. To perform this test, we generate structurally equivalent source and destination XML schemas and input XML documents which satisfy them. XML input document size is directly proportional to schema size; with 2047 schema elements, the input document is 176KBytes, while using 4095 elements a source document is 378KBytes. Figure 8 shows

[4]http://xmlxslt.sourceforge.net/
[5]http://xml.apache.org/xalan-j/
[6]http://4suite.org/
[7]although Python is interpreted, the 4Suite library is statically linked to natively compiled code

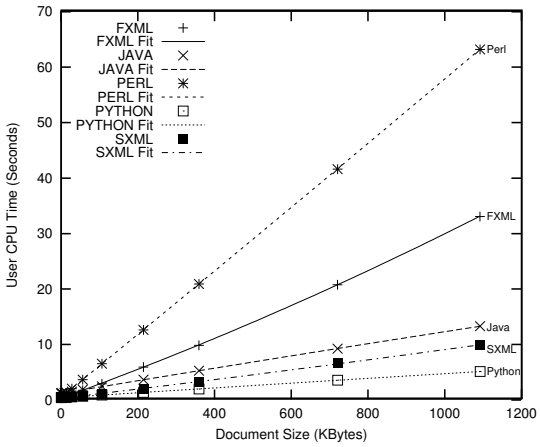


Figure 7. Transformation Performance against increasing XML Document Size

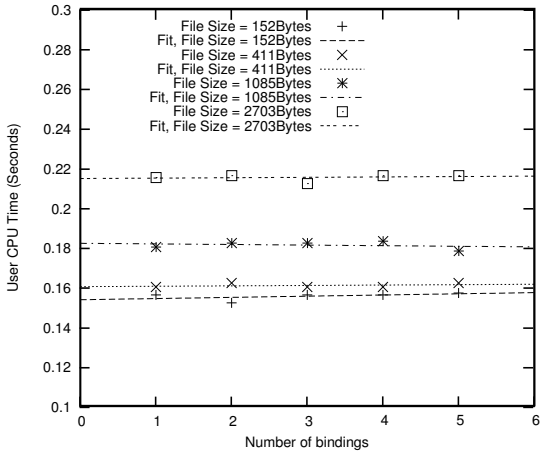


Figure 9. Transformation Performance against number of bindings

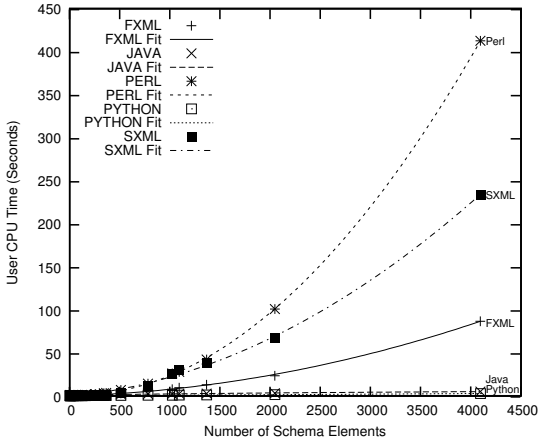


Figure 8. Transformation Performance against increasing XML Schema Size

translation time against the number of schema elements used. Python and JAVA perform the best - a linear expansion with respect to schema size that remains very low in comparison to FXML-T and Perl. FXML-T itself has a quadratic expansion; however, upon further examination, we find the quadratic expansion emanates from the XML parsing subroutines used to read schemas and M-Bindings, whereas the translation itself has a cost linear to the size of its input. The SCHEME XML library used for XML parsing is common to FXML-T and SXML, hence the quadratic expansion for SXML also. Therefore, our translation approach is linear when implemented with a suitable XML parser.

5.2. Composition Cost

Binding composition comes with virtually no performance cost.

One important feature of our translation language (FXML) is the ability to compose M-Bindings at runtime. This can achieved by creating an M-Binding that includes individual mappings from an external M-Binding, or imports all mappings from an external M-binding. For Service interfaces operating over multiple schemas, M-Bindings can be composed easily from existing Type Adaptors. Ideally, this composability should come with minimal cost. To examine M-Binding cost, we increased the number of M-Bindings imported and observed the time required to transform the document. To perform the translation, 10 mappings are required $m_1, m_2, \ldots, m_{10}$. M-Binding 1 contains all the required mapping statements: $B_1 = \{m_1, m_2, \ldots, m_{10}\}$. M-Binding 2 is a composition of two M-Bindings where $B_2 = \{m_1, \ldots, m_5\} \cup B_{2a}$ and $B_{2a} = \{m_6, \ldots, m_{10}\}$. To fully test the cost of composition, we increased the number of M-Bindings used and run each test using 4 source documents with sizes 152Bytes, 411Bytes, 1085Bytes, and 2703Bytes. While we aim for zero composability cost, we would expect a small increase in translation time as more M-Bindings are included. By increasing source document size, a larger proportion of the translation time will be spent on reading in the document and translating it. Consequently, the relative cost of composing M-Bindings will be greater for smaller documents and therefore the increase in cost should be greater. Figure 9 shows the time taken to transform the same four source documents against the same mappings distributed across an increasing number of M-Bindings. On the whole, a very subtle increase in performance cost is seen, with the exception of the file size

1085Bytes. We attribute this anomaly to the rate of error in time recording which is only accurate to ± 10 milliseconds.

5.3. Registry Cost

The cost of M-Binding discovery using Grimoires is not significant when compared to cost of executing the target service.

Our final test examines the cost of using Grimoires to discover the required M-Bindings. While this feature facilitates automation, it will require additional Web Service invocations in a workflow.

Activity	*Average*
DDBJ Execution	2.50
Realisation Discovery	0.22
Realisation Transformation	0.47
Jena Mediation	0.62
Serialisation Discovery	0.23
Serialisation Translation	0.27
Total Mediation	1.81

The Table above shows the average time taken (from 5 runs) for a type translation using OWL as an intermediarry representation. These types of translation consist of 5 steps: (i) discover realisation M-Binding, (ii) transformation of result to OWL instance, (iii) import OWL instance to JENA KB, (iv) discovery of serialisation M-Binding, (v) trasformation of OWL instance to XML. Results show that the total mediation time is roughly 2 seconds, with the largest portion of the time taken importing the OWL instance into JENA. The discovery overhead (realisation and serialisation M-Bindings) is acceptable in comparison to service execution and translation times.

6. Related Work

The Interoperability and Reusability of Internet Services (IRIS) project have also recognised the need to assist bioinformaticians in the design and execution of Workflows. Radetzki *et al* [16] describe adaptor services using WSDL with additional profiles in the Mediator Profile Language (MPL)[8]. adaptor services are collated in a *Mediator Pool* and queried using a derivation from the input and output descriptions of services connected by a dataflow. The query is a combination of syntactic information and semantic concepts from a domain ontology. A matchmaking algorithm presents a ranked list of adaptors, some of which may be composed from multiple adaptor instances. To aid the Adaptor description stage, WSDL descriptions of services and their documentation are linguistically analised to establish the sort of service. For example, the 'getNucSeq' method is decomposed into the terms 'get', 'nuc', and 'seq'. To this end, the matching algorithm is relaxed and not intended to be used automatically; instead, users are aided during workflow composition.

Within the SEEK framework [4], each service has a number of ports which expose a given functionality. Each port advertises a *structural type* which defines the input and output data format by a reference to an XML schema type. If the output of one service port is used as input to another service port, it is defined as *structurally valid* when the two types are equal. Each service port can also be allocated a *semantic type* which is specified by a reference to a concept within an ontology. If two service ports are plugged together, they are *semantically valid* if the output from the first port is subsumed by the input to the second port. Structural types are linked to semantic types by a registration mapping using a custom mapping language based on XPATH. If the concatenation of two ports is semantically valid, but not structurally valid, an XQUERY transformation can be generated to integrate the two ports, making the link *structurally feasible*. The SEEK system provides data integration between different logical organisations of data using a common conceptual representation, the same technique that we adopt. However, their work is only applicable to services within the bespoke SEEK framework. The architecture we present is designed to work with arbitrary WSDL Web Services annotated using conventional semantic Web Service techniques.

Hull *et al* [8] dictate that conversion services, or *shims*, can be placed in between service whenever some type of translation is required - exactly as the current MYGRID solution. They explicitly specify that a shim service is *experimentally neutral* in the sense that it has no side-effect on the result of the experiment. By enumerating the types of shims required in bioinformatics Grids and classifying all instances of shim services, it is hoped that the necessary translation components could be automatically inserted into a workflow. However, their focus is not on the translation between different data representation, rather the need to manipulate data sets; extracting information from records, finding alternative sources for data, and modifying workflow designs to cope with iterations over data sets.

Moreau *et al* [15], have investigated the same problem within the Grid Physics Network, GriPhyn[9]. To provide a homogeneous access model to varying data sources, Moreau *et al* propose a separation between logical and physical file structures. This allows access to data sources to be expressed in terms of the logical structure of the information, rather than the way it is physically represented. The XML Data Type and Mapping for Specifying Datasets (XDTM) prototype provides an implementation which allows data source to be navigated using XPATH. While this approach is useful when amalgamating data from different

[8] http://www.cs.uni-bonn.de/III/bio/iris/MediatorProfile.owl

[9] http://griphyn.org/

physical representations (i.e. XML files, binary files and directory structures), it does not address the problem of data represented using different logical representations (i.e. different schemas with the same physical representation). Our service integration problem arises from the fact that different service providers use different logical representations of conceptually equivalent information.

7. Conclusions and Future Work

In this paper, we have described the motivation for a generic Type Adaptor description policy to support automated workflow harmonisation when syntactic incompatibilities are encountered. By using WSDL to describe Type Adaptor capabilities, and the Grimoires registry to advertise them, translation components can be discovered at runtime and placed into the running workflow. Using FXML-M as a Type Adaptor language and OWL ontologies as an intermediate representation, we show an implementation that supports a bioinformatics use case. Evaluation of our binding language approach, its implementation, and the use of Grimoires as a registry shows our architecture is scalable, supports Type Adaptor composablity with virtually no cost, and has a relatively low overhead for binding discovery.

While our Architecture has been based on Type Adaptors taking the form of FXML-M Binding documents, the approach will work with any Type Adaptor language, such as XSLT and JAVA, providing the appropriate code is put in place to automatically generate WSDL descriptions of their capabilities and the workflow engine is able to interpret their WSDL binding (i.e. execute an XSLT script, run a JAVA program, invoke a particular SOAP service). Even though our Architecture has been designed to fit within the existing MYGRID application, this approach will apply to any Grid or Web Services architecture. When incorporating the use of Semantic Web technology, namely the association of WSDL message parts with concepts from an ontology, we have followed existing practices such as those used by the FETA system, OWL-S [13], WSMO [17], and WSDL-S [2].

In future work, we aim to improve the FXML-T implementation in the area of XML parsing. This will make FXML-T translation cost scale at an acceptable rate when increasing both XML document size and XML schema size.

8. Acknowledgment

This research is funded in part by EPSRC myGrid project (reference GR/R67743/01).

References

[1] UDDI technical white paper, September 2000.

[2] R. Akkiraju, J. Farrell, J.Miller, M. Nagarajan, M. Schmidt, and A. S. K. Verma. Web service semantics - WSDL-S. Technical report, UGA-IBM, 2005.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, pages 34 – 43, 2001.

[4] S. Bowers and B. Ludascher. An ontology-driven framework for data transformation in scientific workflows. In *Intl. Workshop on Data Integration in the Life Sciences (DILS'04)*, 2004.

[5] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web services description language (WSDL) 1.1, March 2001. W3C.

[6] J. Clark. XSL transformations (XSLT) version 1.0. Technical report, W3C, 1999.

[7] C. Goble, S. Pettifer, R. Stevens, and C. Greenhalgh. Knowledge Integration: In silico Experiments in Bioinformatics. In I. Foster and C. Kesselman, editors, *The Grid: Blueprint for a New Computing Infrastructure Second Edition*. Morgan Kaufmann, November 2003.

[8] D. Hull, R. Stevens, and P. Lord. Describing web services for user-oriented retrieval. 2005.

[9] N. Kavantzas, D. Burdet, G. Ritzinger, and T. Fletcher. WS-CDL web services choreography description language version 1.0. Technical report, W3C, 2004.

[10] G. Klyne and J. J. Carroll. Resource description framework (RDF): Concepts and abstract syntax. Technical report, W3C, 2004.

[11] F. Leymann. Web services flow language (WSFL 1.0), May 2001.

[12] P. Lord, P. Alper, C. Wroe, and C. Goble. Feta: A lightweight architecture for user oriented semantic service discovery. In *The Semantic Web: Research and Applications: Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece*, pages 17 – 31, Jan. 2005.

[13] D. Martin, M. Burstein, G. Denker, J. Hobbs, L. Kagal, O. Lassila, D. McDermott, S. McIlraith, M. Paolucci, B. Parsia, T. Payne, M. Sabou, E. Sirin, M. Solanki, N. Srinivasan, and K. Sycara. OWL-S: Semantic markup for web service. Technical report, The OWL Services Coalition, 2003.

[14] S. Miles, J. Papay, T. Payne, M. Luck, and L. Moreau. Towards a protocol for the attachment of metadata to service descriptions and its use in semantic discovery. *Scientific Programming*, pages 201–211, 2005.

[15] L. Moreau, Y. Zhao, I. Foster, J. Voeckler, and M. Wilde. XDTM: the XML Dataset Typing and Mapping for Specifying Datasets. In *Proceedings of the 2005 European Grid Conference (EGC'05)*, Amsterdam, Nederlands, Feb. 2005.

[16] U. Radetzki, U. Leser, S. Schulze-Rauschenbach, J. Zimmermann, J. Lussem, T. Bode, and A. Cremers. Adapters, shims, and glue - service interoperability for in silico experiments. *Bioinformatics*, 22(9):1137–1143, 2006.

[17] D. Roman, H. Lausen, and U. Keller. D2v1.0. web service modeling ontology (WSMO), September 2004. WSMO Working Draft.

[18] M. Szomszor, T. R. Payne, and L. Moreau. Automatic syntactic mediation for web service integration. In *Proceedings of the International Conference on Web Services 2006 (ICWS2006)*, Chicago, USA, Sept. 2006.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Grid Workflow Scheduling In WOSE

Yash Patel [1], Andrew Stephen M^{c}Gough [2], John Darlington [3]

London e-Science Centre, Department of Computing, Imperial College
South Kensington Campus, London SW7 2AZ, United Kingdom
{yp03 [1], asm [2], jd [3]}@doc.ic.ac.uk

***Abstract*— The success of web services has infuenced the way in which grid applications are being written. Grid users seek to use combinations of web services to perform the overall task they need to achieve. In general this can be seen as a set of services with a workflow document describing how these services should be combined. The user may also have certain constraints on the workflow operations, such as execution time or cost to the user, specified in the form of a Quality of Service (QoS) document. These workflows need to be mapped to a subset of the Grid services taking the QoS and state of the Grid into account – service availability and performance. We propose in this paper an approach for generating constraint equations describing the workflow, the QoS requirements and the state of the Grid. This set of equations may be solved using Integer Linear Programming (ILP), which is the traditional method. We further develop a 2-stage stochastic ILP which is capable of dealing with the volatile nature of the Grid and adapting the selection of the services during the life of the workflow. We present experimental results comparing our approaches, showing that the 2-stage stochastic programming approach performs consistently better than other traditional approaches. This work forms the workflow scheduling service within WOSE (Workflow Optimisation Services for e-Science Applications), which is a collaborative work between Imperial College, Cardiff University and Daresbury Laborartory.**

I. Introduction

Grid Computing has been evolving over recent years towards the use of service orientated architectures [1]. Functionality within the Grid exposes itself through a service interface which may be a standard web service endpoint. This functionality may be exposing computational power, storage, software capable of being deployed, access to instruments or sensors, or potentially a combination of the above.

Grid workflows that users write and submit may be abstract in nature, in which case the final selection of web services has not been finalised. We refer to the abstract description of services as abstract services in this paper. Once the web services are discovered and selected, the workflow becomes concrete, meaning the web services matching the abstract description of services are selected.

The Grid is by nature volatile – services appear and disappear due to changes in owners policies, equipment crashing or network partitioning. Thus submitting an abstract workflow allows late binding of the workflow with web services currently available within the Grid. The workflow may also take advantage of new web services which were not available at the time of writing. Users who submit a workflow to the Grid will often have constraints on how they wish the workflow to perform. These may be described in the form of a QoS document which details the level of service they require from the Grid. This may include requirements on such things as the overall execution time for their workflow; the time at which certain parts of the workflow must be completed; and the cost of using services within the Grid to complete the workflow.

In order to determine if these QoS constraints can be satisfied it is necessary to store historic information and monitor performance of different web services within the Grid. Such information could be performance data related to execution and periodic information such as queue length, availability. Here we see that existing Grid middleware for performance repositories may be used for the storage and retrieval of this data. If the whole of the workflow is made concrete at the outset, it may lead to QoS violations. Therefore we have adopted an iterative approach. At each stage the workflow is divided into those abstract services which need to be deployed now and those that can be deployed later. Those abstract services which need to be deployed now are made concrete and deployed to the Grid. However, to maintain QoS constraints it is necessary to ensure that at each iteration the selected web services will still allow the whole workflow to achieve QoS.

This paper presents results of the workflow scheduling service within WOSE (Workflow Optimisation Services for e-Science Applications). WOSE is an EPSRC-funded project jointly conducted by researchers at Imperial College, Cardiff University and Daresbury Laboratory. We discuss how our work relates to others in the field in Section II. Section III describes the process of workflow aware performance guided scheduling, followed by a description of the 2-stage stochastic programming approach and an algorithm for stochastic scheduling in Section IV. In Section V we illustrate how our approach performs through simulation before concluding in Section VI.

II. Related Work

Business Process Execution Language (BPEL) [2] is beginning to become a standard for composing web-services and many projects such as Triana [3] and WOSE [4] have adopted it as a means to realise service-based Grid workflow technology. These projects provide tools to specify abstract workflows and workflow engines to enact workflows. Buyya et al [5] propose a Grid Architecture for Computational Economy

TABLE I
SCHEDULING PARAMETERS.

Symbol	Name
A_i	Abstract service i
	Expected time, cost and selection
a_{ir}, c_{ir},	variable associated with r^{th}
x_{ir}	web service matching A_i
$time_{QoS}$	Maximum time in which the workflow should get executed
$deadline_i$	Time in which A_i is expected to complete
$\|A\|$	Number of abstract services
	Number of web services
$\|a_i\|$	matching A_i

(GRACE) considering a generic way to map economic models into a distributed system architecture. The Grid resource broker (Nimrod-G) supports deadline and budget based scheduling of Grid resources. However no QoS guarantee is provided by the Grid resource broker. Zeng et al [6] investigate QoS-aware composition of Web Services using integer programming method. The services are scheduled using local planning, global planning and integer programming approaches. The execution time prediction of web services is calculated using an arithmetic mean of the historical invocations. However Zeng et al assume that services provide upto date QoS and execution information based on which the scheduler can obtain a service level agreement with the web service. Brandic et al [7] extend the approach of Zeng et al to consider application-specific performance models. However their approach fails to guarantee QoS over entire life-time of a workflow. They also assume that web services are QoS-aware and therefore certain level of performance is guaranteed. However in an uncertain Grid environment, QoS may be violated. Brandic et al have no notion of global planning of a workflow. Thus there is a risk of QoS violation. Huang et al [8] have developed a framework for dynamic web service selection for the WOSE project. However it is limited only to best service selection and no QoS issues are considered. We see our work fitting in well within their optimisation service of the WOSE architecture. A full description of the architecture can be found in [8]. Our approach not only takes care of dynamically selecting the optimal web service but also makes sure that overall QoS requirements of a workflow is satisfied with sufficiently high probability. The main contribution of our paper is the novel QoS support approach and an algorithm for stochastic scheduling of workflows in a volatile Grid.

III. WORKFLOW AWARE PERFORMANCE GUIDED SCHEDULING

We provide Table: I as a quick reference to the parameters of the ILP.

A. Deterministic Integer Linear Program (ILP)

Before presenting our 2-stage stochastic integer linear program we first present the deterministic ILP program. The program is integer linear as it contains only integer variables (unknowns) and the constraints appearing in the program are all linear. The ILP consists of an objective which we wish to minimise along with several constraints which need to be satisfied. The objective here is to minimise the overall workflow cost:

$$Cost = minimize[O] \tag{1}$$

$$O = \sum_{i}^{|A|} \sum_{r}^{|a_i|} c_{ir} x_{ir} \tag{2}$$

O is the cost associated with web services. We have identified the following constraints.

- **Selection Constraint :**

$$\forall i, \sum_{r}^{|a_i|} x_{ir} = 1 \tag{3}$$

$$x_{ir} \in \{0, 1\} \tag{4}$$

  Equation 3 takes care of mapping A_i to one and only one web service. For each A_i, only one of the x_{ir} equals 1, while all the rest are 0.
- **Deadline Constraint :** Equation 5 ensures that A_i finishes within the assigned deadline.

$$\sum_{r}^{|a_i|} a_{ir} x_{ir} \leq deadline_i \tag{5}$$

- **Other workflow specific constraints :** These constraints are generated based on the workflow nature and other soft deadlines (execution constraints). This could be explicitly specified by the end-user. e.g. some abstract service or a subset of abstract services is required to be completed within t seconds. These could also be satisfying other QoS parameters such as reliability and availability. A full list of constraints is beyond the scope of this paper.

IV. TWO-STAGE STOCHASTIC ILP WITH RECOURSE

Stochastic programming, as the name implies, is mathematical (i.e. linear, integer, mixed-integer, nonlinear) programming but with a stochastic element present in the data. By this we mean that in deterministic mathematical programming the data (coefficients) are known numbers while in stochastic programming these numbers are unknown, instead we may have a probability distribution present. However these unknowns, having a known distribution could be used to generate a finite number of deterministic programs through techniques such as Sample Average Approximation (SAA) and an ϵ-optimal solution to the true problem could be obtained. A full discussion of SAA is beyond the scope of this paper and interested readers may refer [9].

Consider a set S of abstract services that can be scheduled currently and concurrently. Let $|S|$ be the number of such services. Similarly let P be the set of unscheduled abstract services and $|P|$ be its number. Equations (6) to (9) represent a 2-stage stochastic program with recourse, where stage-1 minimises current costs and stage-2 aims to minimise future costs. The recourse term is $Q(x_S, \omega)$, which is the future cost. The term $e^T z$ in the objective of the stage-2 program is the

penalty incurred for failing to compute a feasible schedule. The vector e has values such that the incurred penalty is clearly apparent in the objective value. The z variables are also present in the constraints of stage-2 programs in order to keep the program feasible as certain realisations of random variables will make the program infeasible. The vector z consists of continuous variables whose size depends on the number of constraints appearing in the program.

$$Cost = minimise[O + E(Q(x_S, \omega))] \quad (6)$$

- **Stage-1**

$$O = \sum_{i}^{|S|} \sum_{r}^{|a_i|} c_{ir} x_{ir} \quad (7)$$

Subject to the following constraints: selection, scheduling along with other possible constraints.

- **Stage-2**

ω is a vector consisting of random variables of runtimes and costs of services. x_S is the vector denoting the solutions of stage-1. Q(x_S,ω) is the optimal solution of

$$Cost_\xi = minimise[\xi] + \mathbf{e^T z} \quad (8)$$

$$\xi = \sum_{i}^{|P|} \sum_{r}^{|a_i|} c_{ir} x_{ir} \quad (9)$$

Subject to the following constraints: selection, scheduling along with other possible constraints. ξ is a realisation of expected costs of using services. The function E is the expected objective value of stage-2, which is computed using the SAA problem listed in equation (10). The stage-2 solution can be used to recompute stage-1 solution, which in turn leads to better stage-2 solutions.

$$minimise[O + \frac{1}{N} \sum_{n=1}^{N} Q(x_S, \xi^n)] \quad (10)$$

$$N \geq \frac{3\sigma_{max}^2}{(\epsilon - \delta)^2} \frac{log|F|}{\alpha} \quad (11)$$

In equation (11), $|F|$ is the number of elements in the feasible set, which is the set of possible mappings of abstract services to real Grid services. 1 - α is the desired probability of accuracy, δ the tolerance, ϵ the distance of solution to true solution and σ_{max}^2 is the maximum execution time variance of a particular service in the Grid. One could argue that it may not be trivial to calculate both σ_{max}^2 and $|F|$. Maximum execution time variance of some Grid service could be a good approximation for σ_{max}^2 and $|F|$ could be obtained with proper discretisation techniques. Equation (11) is derived in [10]. Our scheduling service provides a 95% guarantee. Hence 1 - α is taken as 0.95. ϵ - δ is taken as 2 for convenience, while $log|F|$ turns out to be approximately equal to 4. In our case in order to obtain 95% confidence level, N approximately turns out to be around 600. This means that one needs to solve nearly 600 deterministic ILP programs in stage-2 for each iteration of algorithm 1. The number of unknowns in the ILP being only about 500, negligible time is spent to solve these many scenarios.

A. Algorithm for stochastic scheduling of workflows

Algorithm 1 obtains scheduling solutions for abstract workflow services by solving 2-stage stochastic programs, where stage-1 minimises current costs and stage-2 minimises future costs. This algorithm guarantees an ϵ-optimal solution (i.e., a solution with an absolute optimality gap of ϵ to the true solution) with desired probability [9]. However to achieve the desired accuracy one needs to sample enough scenarios, which often get quite big in a large utility grid, and in a service rich environment with continuous execution time distributions associated with Grid services, the number of scenarios is theoretically infinite. However with proper discretisation techniques the number of scenarios or the sample size required to get the desired accuracy is at most linear in the number of Grid services. This is clearly evident from the value of N (equation (11)), which is the sample size, as $|F|$ being the size of feasible set, is exponential in the number of Grid services. Finally statistical confidence intervals are then derived on the quality of the approximate solutions.

Algorithm 1 initially obtains scheduling solutions for stage-1 abstract services, S in the workflow. This stage-1 result puts constraints on stage-2 programs, which aims at finding scheduling solutions for rest of the unscheduled workflow. The sampling size (equation (11)) for each iteration, guarantees an ϵ-optimal solution to the true scheduling problem with desired accuracy, 95% in our case. If the optimality gap or variance of the gap estimator are small, only then the scheduling operation is a success. If not, the iteration is repeated as mentioned in step 3.6 of the algorithm. This leads to computing new schedule for stage-1 abstract services with tighter QoS bounds. When the scheduled stage-1 abstract services finish execution, algorithm 1 is used to schedule abstract services that follow them in the workflow. Step 4 selects the stage-1 solution, which has a specified tolerance δ to the true problem with probability at least equal to specified confidence level 1 - α.

$$L = \frac{\sum_{m=1}^{M} O^m}{M} \quad (12)$$

$$Var^L = \frac{\sum_{m=1}^{M} (O^m - L)^2}{M(M-1)} \quad (13)$$

$$U = O_1 + \frac{1}{N'} \sum_{n=1}^{N'} Q(x_S, \xi^n) \quad (14)$$

$$Var^U = \frac{\sum_{n=1}^{N'} (Q(x_S, y_S, \xi^n) - U)^2}{N'(N'-1)} \quad (15)$$

Algorithm 1 initially obtains scheduling solutions for stage-1 abstract services, S in the workflow. This stage-1 result puts constraints on stage-2 programs, which aims at finding scheduling solutions for rest of the unscheduled workflow. The sampling size (eq. 11) for each iteration, guarantees an ϵ-optimal solution to the true scheduling problem with desired accuracy, 95% in our case. If the optimality gap or variance of the gap estimator are small, only then the scheduling operation is a success. If not, the iteration is repeated as mentioned in step 3.6 of the algorithm. This leads to computing new

Algorithm 1 Algorithm for stochastic scheduling

Step 1 : Choose sample sizes N and $N^{'} \geq N$, iteration count M, tolerance ϵ and rule to terminate iterations
Step 2 : Check if termination is required
for m = 1, . . .,M **do**
Step 3.1 : Generate N samples, and solve the SAA problem, let the optimal objective be O^m for corresponding iteration
end for
Step 3.2 : Compute a lower bound estimate L (eq. 12) on the objective and its variance Var^L (eq. 13)
Step 3.3 : Generate $N^{'}$ samples, use one of the feasible stage-1 solution and solve the SAA problem to compute an upper bound estimate U (eq. 14) on the objective and its variance Var^U (eq. 15)
Step 3.4 : Estimate the optimality gap ($Gap = |L - U|$) and the variance of the gap estimator ($Var^{Gap} = Var^L + Var^U$)
Step 3.5 : If Gap and Var^{Gap} are small, choose stage-1 solution. Stop
Step 3.6 : If Gap and Var^{Gap} are large, tighten stage-1 QoS bounds, increase N and/or $N^{'}$, goto **step 2**

schedule for stage-1 abstract services with tighter QoS bounds. Step 3.5 selects the stage-1 solution, which has a specified tolerance δ to the true problem with probability at least equal to specified confidence level 1 - α.

V. Experimental Evaluation

In this section we present experimental results for the ILP techniques described in this paper.

A. Setup

Table II summarises the experimental setup. We have performed 3 simulations and for each different setup of a simulation we have performed 10 runs and averaged out the results. Initially 500 jobs allow the system to reach steady state, the next 1000 jobs are used for calculating statistics such as mean execution time, mean cost, mean failures, mean partial executions and mean utilisation. The last 500 jobs mark the ending period of the simulation. Mean of an abstract service is measured in millions of instructions (MI). In order to compute expected runtimes, we put no restriction on the nature of execution time distribution and apply Chebyshev inequality [11] to compute expected runtimes such that 95% of jobs would execute in time under a_{ir} (equation (16)). It should be noted that such bounds or confidence intervals on the execution times can also be computed using other techniques such as Monte Carlo approach [12] and Central Limit Theorem [11] or by performing finite integration, if the underlying execution time PDFs (Probability Density Functions) are available in analytical forms. The waiting time is also computed in such a way that in 95% of the cases, the waiting time encountered will be less than the computed one. The value 4.47 appearing in the equations below is due to applying Chebyshev inequality for including 95% of the execution or waiting time distribution area. In equation (16), μ_{ir} and σ_{ir} are the mean and standard deviation of the execution time distribution of a running software service. c_{ir} is a simple product function of a_{ir}.

$$a_{ir} = \mu_{ir} + 4.47\sigma_{ir} + waiting\ time \tag{16}$$

a_{ir} (equation (17)) for stage-2 programs is calculated in a slightly different fashion.

$$a_{ir} = \xi^e(\mu_{ir}, \sigma_{ir}^2) + \xi^w(\mu_r, \sigma_r^2) \tag{17}$$

Here ξ^e is the execution time distribution sample of an abstract service on a Grid service. ξ^w is the waiting time distribution sample associated with R_r. We have used Monte-Carlo [12] technique for sampling values out of the distributions. Other sampling techniques such as Latin Hypercube sampling could also be used in place. We provide an example for calculating initial deadlines, given by equation (18) for the first abstract service (generate matrix) of workflow type 1. Deadline calculation of an abstract service takes care of all possible paths in a workflow and scaling is performed with reference to the longest execution path in a workflow. Equation (18) is scaled with reference to $time_{QoS}$. It should be noted that initially implies calculation before performing the iterations of algorithm 1. Subsequent deadlines of abstract services in a workflow are calculated initially by scaling with reference to the remaining workflow deadline.

$$deadline_1 = \frac{X_1}{X_1 + X_{234}} time_{QoS} \tag{18}$$

$$X_1 = \mu_i^{max}(1 + 4.47CV_i^{max}) \tag{19}$$

$$X_{234} = \sum_{j=2}^{4} \mu_j^{max}(1 + 4.47CV_j^{max}) \tag{20}$$

Initial deadline calculation is done in order to reach an optimal solution faster. We are currently investigating cut techniques which can help to reach optimal solutions even faster. Here μ_i^{max} and CV_i^{max} are the mean and coefficient of variation of a Grid service that has the maximum expected runtime. If Gap and Var^{Gap} are large, bounds are tightened in such a way that in the next iteration they become smaller. e.g. minimum coefficient for time (a_{ir}) could be set as the deadline or recourse term variable values (z) in the stage-2 programs could be used to tighten deadline. The workflows experimented with are shown in figure 1. The workflows are simulation counterparts of the real world workflows. Their actual execution is a delay based on their execution time distribution, as specified in table II. In the first simulation, type 1 workflows are used, in the second simulation, type 2 workflows are used and in the third simulation workload is made heterogenous (HW). The type 1 workflow is quite simple compared to type 2, which is a real scientific workflow. All the workflows have different QoS requirements as specified in table II. The ILP solver used is CPLEX by ILOG [13], which is one of the best industrial quality optimisation software. The simulation is developed on top of simjava 2 [14], a discrete event simulation package. The Grid size is kept small in order to get an asymptotic

TABLE II

SIMULATION PARAMETERS.

Simulation	**1**	**2**	**3**
Services matching A_i	24	12	24
Service speed (kMIPS)	3-14	3-14	3-14
Unit cost (per sec)	5-29	5-29	5-29
Arrival Rate (λ) (per sec)	1.5-10	0.1-2.0	1.5-3.6
A_i Mean (μ) (kMI)	7.5-35	10-30	7.5-35
A_i CV = σ/μ	0.2-2.0	0.2-1.4	0.2-2.0
Workflows	Type 1	Type 2	HW
$time_{QoS}$ (sec)	40-60	80-100	40-60

behaviour of workflow failures as coefficient of variation (CV) of execution or arrival rates (λ) are increased.

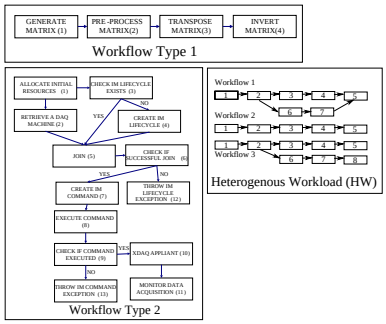


Fig. 1. Workflows

B. Results

We compare our scheme (DSLC) with two traditional schemes (DDLC and SDLC), all with a common objective of minimising cost and ensuring workflows execute within deadlines. The workflows don't have any slack period, meaning they are scheduled without any delay as soon as they are submitted. DDLC (dynamic, deterministic, least cost satisfying deadlines) and DSLC (dynamic, stochastic, least cost satisfying deadlines) job dispatching strategies calculate an initial deadline based on equation (18). Though DDLC calculates new deadlines each time it needs to schedule abstract services, the deadlines don't change once they are calculated. The deadlines get changed iteratively in case of DSLC due to the iterative nature of algorithm 1. Scheduling of abstract services continues until the lifetime of workflows in case of DDLC and DSLC. It is not the case with SDLC (static, deterministic, least cost satisfying deadlines) and as soon as the workflows are submitted, an ILP is solved and scheduling solutions for all abstract services within the workflows are obtained. In case of SDLC, once the scheduling solutions are obtained, they don't get changed during the entire lifetime of the workflows. The main comparison metrics here are mean cost, mean time, failures and mean utilisation as we increase λ and CV. However we will keep our discussion limited to failures as a workflow failure means failure in satisfying QoS requirements of workflows.

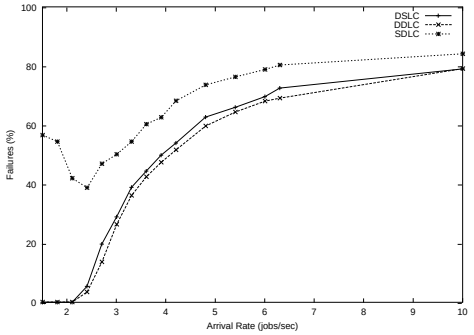


Fig. 2. Failures vs λ, CV = 0.2 (Simulation 1)

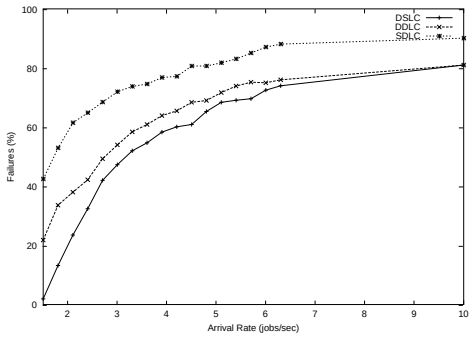


Fig. 3. Failures vs λ, CV = 1.8 (Simulation 1)

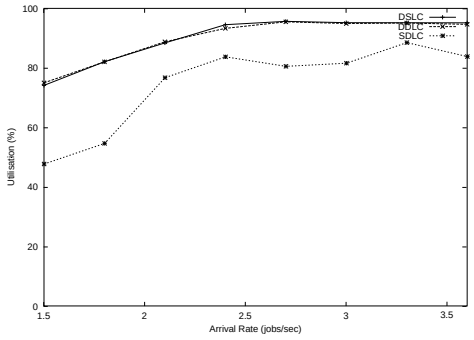


Fig. 4. Avg Utilisation vs λ, CV = 0.2 (Simulation 1)

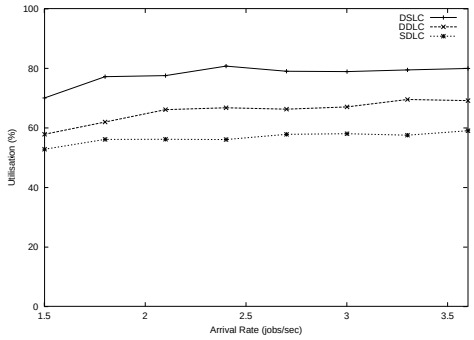


Fig. 5. Avg Utilisation vs λ, CV = 1.8 (Simulation 1)

C. Effect of arrival rate and workload

We see that in case of figures 2 and 3, as λ increases, DSLC continues to outperform other schemes. This trends continues however but with a reduced advantage. This can be explained as follows. This trends continues however but the

advantage keeps on reducing as arrival rates increase. This can be explained as follows. When arrival rates increase, more work needs to be scheduled in the same amount of time, as previously available. Moreover it is safe to assume that response time of services is an increasing function of arrival rate. Hence failures increase. Moreover this behaviour not being linear and failures themselves reaching a limiting value, this advantage is reduced. SDLC obtains a joint solution and therefore is a sub-optimal solution or is optimal only at the time of scheduling. Hence more failures are registered in case of SDLC. Referring to figures 4 and 5, it is apparent that when CV is low, utilisation in case of DSLC and DDLC turns out to be the same. However SDLC also registers reasonable utilisation. Overall utilisation is maximum in case of DSLC due to its capability of obtaining optimal solutions. When CV is high, DSLC still outperforms other schemes. Due to high unpredicatability, DDLC and SDLC register moderate utilisations. In case of workflow type 2, for low and high CVs, as λ is increased, DSLC outperforms all other schemes. In case of utilisation, for low CV, all schemes register high utilisations. However in case of high CV, DSLC registers far higher utilisation than other schemes. Referring to figures 12 and 13, again DSLC registers lowest failures for both low and high CVs. This is because workload is quite heterogenous and environment therefore becomes quite unpredictable. In this case DSLC obtains better scheduling solutions than other schemes. In case of utilisation (figures 14 and 15), again due to less failures in case of DSLC, utilisation is registered higher than other schemes.

D. Effect of CV

We see that in case of workflow type 1, which is quite predictable and sequential, as arrival rates increase, for low CV (predictable behaviour), DDLC performs slightly better than DSLC. This is because even if DSLC iteratively tightens deadlines, it doesn't help to get a better schedule due to highly predictable environment and as a result failures increase slightly as it tries to schedule workflows which would have failed in case of DDLC. As CV is increased, we see that DSLC outperforms other schemes. This is because the environment becomes less predictable and algorithm 1 obtains better deadline solutions solutions that help to reduce failures. DDLC closes the gap asymptotically as λ increases. This is because failures increase as λ increases and theoretically the workflows themselves cannot be scheduled as they would fail to meet their deadlines. In case of workflow type 2, for both low and high CVs, DSLC performs significantly better than other schemes. Referring to figures 10 and 11, we see that as CV is increased for low arrival rates, utilisation drops which indicates that failures increase, which in turn indicates that environment becomes more and more unpredictable. With high workloads, as CV is increased, utilisation drops, but this time DSLC registers highest utilisation. SDLC and DDLC register lower utilisation as they fail to cope with the increasing uncertainty. However for low CV, they all start off from about the same utilisation mark. When workload is made heterogenous, for both low and high CVs, DSLC outperforms other schemes. For high CV, the environment becomes highly uncertain and hence SDLC registers a spiky behaviour in utilisation. This is in agreement considering its static nature of job assignment.

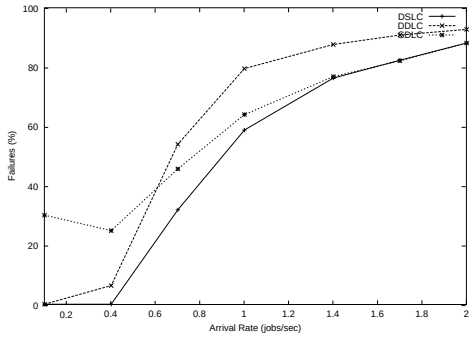


Fig. 6. Failures vs λ, CV = 0.2 (Simulation 2)

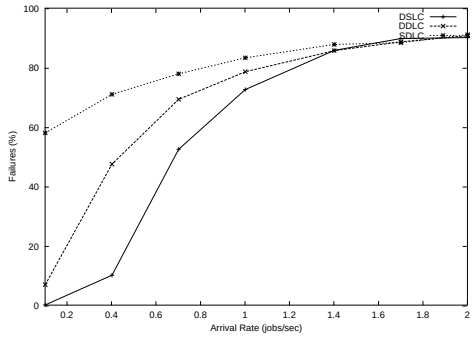


Fig. 7. Failures vs λ, CV = 1.4 (Simulation 2)

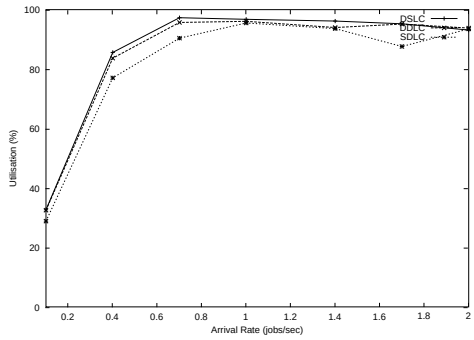


Fig. 8. Avg Utilisation vs λ, CV = 0.2 (Simulation 2)

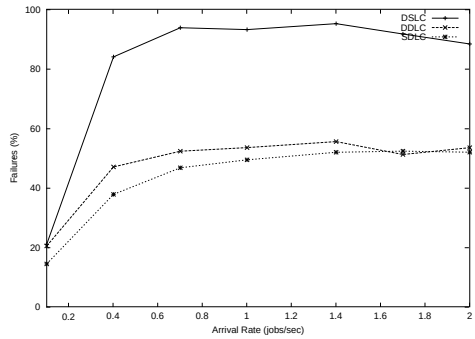


Fig. 9. Avg Utilisation vs λ, CV = 1.4 (Simulation 2)

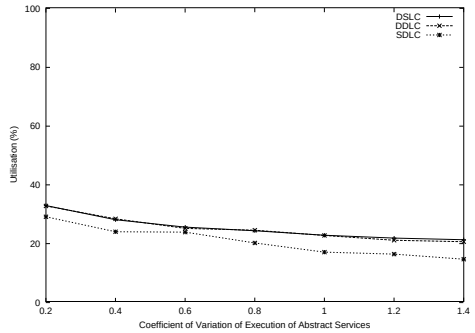


Fig. 10. Avg Utilisation vs CV, λ = 0.1 (Simulation 2)

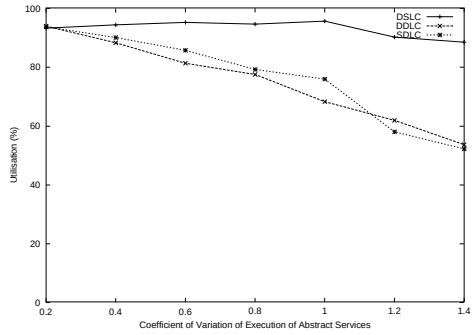


Fig. 11. Avg Utilisation vs CV, λ = 2.0 (Simulation 2)

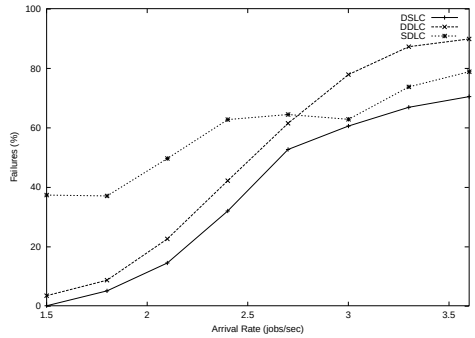


Fig. 12. Failures vs λ, CV = 0.2 (Simulation 3)

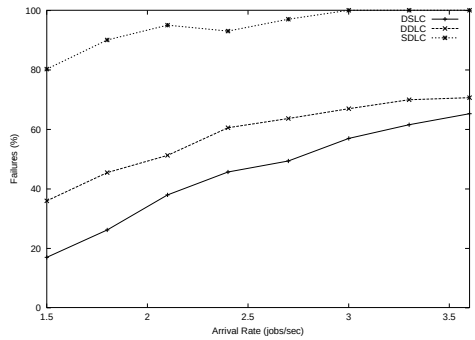


Fig. 13. Failures vs λ, CV = 1.8 (Simulation 3)

E. Effect of workflow nature

Workflow type 2 is more complex and far less predictable than workflow type 1. Hence in such case we see that DSLC outperforms other schemes for low and high CVs. This is to say that DSLC algorithm obtains better deadline solutions by solving the SAA problem than other schemes, as a result of which less failures are experienced. The other schemes, since they obtain static deadlines, fail to outperform DSLC. However when λ increases, all the curves merge to values closer to 100%. In case of heterogenous workload, the environment again becomes less predictable and as a result DSLC continues to outperform other schemes.

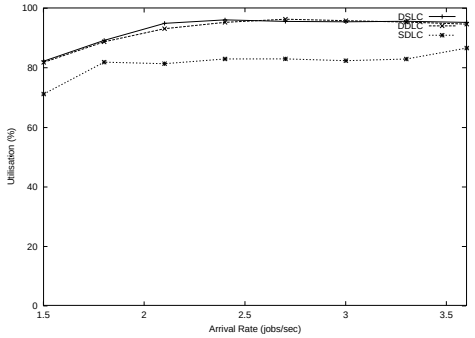


Fig. 14. Avg Utilisation vs λ, CV = 0.2 (Simulation 3)

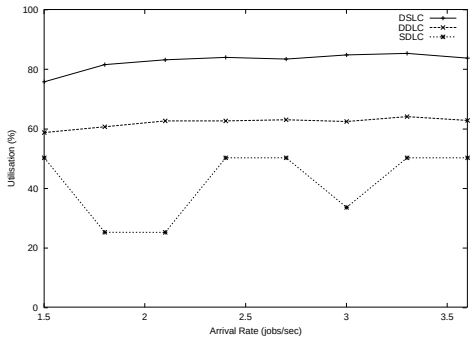


Fig. 15. Avg Utilisation vs λ, CV = 1.8 (Simulation 3)

VI. Conclusion and Future Work

We have developed a 2-stage stochastic programming approach to workflow scheduling using an ILP formulation of QoS constraints, workflow structure, performance models of Grid services and the state of the Grid. The approach gives a considerable improvement over other traditional schemes. This is because SAA approach obtains ϵ-optimal solutions minimised and approximated over uncertain conditions while providing QoS guarantee over the workflow time period. The developed approach performs considerably better particularly when the CV of execution times and the workflow complexity are high. At both low and high arrival rates, the developed approach comfortably outperforms the traditional schemes.

As future work we seek to extend our model of Grid services and the constraints on these. This will enable us to more accurately schedule workflows onto the Grid. As the number of constraints increase along with a greater number of Grid services we see that the solution time of the ILP may become significant. A parallel approach may be used to improve on this situation. We would like to perform experiments with workflows having a slack period, meaning workflows can wait for sometime before getting serviced. We would also like to develop pre-optimisation techniques that would decrease the

unknowns requiring to be solved in the ILP. i.e. prune certain Grid services from the ILP that cannot improve the expectation of its objective.

REFERENCES

[1] N. Furmento, J. Hau, W. Lee, S. Newhouse, and J. Darlington, "Implementations of a Service-Oriented Architecture on top of Jini, JXTA and OGSI," in *Grid Computing: Second European AcrossGrids Conference, AxGrids 2004*, ser. Lecture Notes in Computer Science, vol. 3165, Nicosia, Cyprus, Jan. 2004, pp. 90–99.

[2] *BPEL Specification*, Std. [Online]. Available: http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/2003/

[3] S. Majithiaa, M. S. Shields, I. J. Taylor, and I. Wang, "Triana: A Graphical Web Service Composition and Execution Toolkit," *International Conference on Web Services*, 2004.

[4] "WOSE (Workflow Optimisation Services for e-Science Applications)." [Online]. Available: http://www.wesc.ac.uk/projects/wose/

[5] R. Buyya et al., "Economic Models for Resource Management and Scheduling in Grid Computing," *Concurrency and Computation*, vol. 14, no. 13-15, pp. 1507–1542, 2002.

[6] L. Zeng et al., "QoS-Aware Middleware for Web Services Composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, May 2004.

[7] I. Brandic and S. Benkner and G. Engelbrecht and R. Schmidt, "QoS Support for Time-Critical Grid Workflow Applications," Melbourne, Australia, 2005.

[8] Lican Huang and David W. Walker and Yan Huang and Omer F. Rana, "Dynamic Web Service Selection for Workflow Optimisation ," in *UK e-Science All Hands Meeting*, Nottingham, UK, Sept. 2005.

[9] Kleywegt and A. Shapiro and H. De-Mello, "The sample average approximation method for stochastic discrete optimization," *SIAM Journal of Optimization*, pp. 479–502, 2001.

[10] T. Homem-de-Mello, "Monte Carlo methods for discrete stochastic optimization," *Stochastic Optimization: Algorithms and Applications*, pp. 95–117, 2000.

[11] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 1972.

[12] N. Metropolis and S. Ulam, "The Monte Carlo Method," *Journal of the American Statistical Association*, 1949.

[13] "ILOG." [Online]. Available: http://www.ilog.com/

[14] "SimJava." [Online]. Available: http://www.dcs.ed.ac.uk/home/hase

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Collaborative study of GENIE*fy* Earth System Models using scripted database workflows in a Grid-enabled PSE

A. R. Price[1], Z. Jiao[1], I. I. Voutchkov[1], T. M. Lenton[2], G. Williams[3], D. J. Lunt[3], R. Marsh[4], P. J. Valdes[3], S. J. Cox[1] and the GENIE team

[1]School of Engineering Sciences, University of Southampton, Southampton, UK
[2]School of Environmental Sciences, University of East Anglia, Norwich, UK
[3]School of Geographical Sciences, University of Bristol, Bristol, UK
[4]National Oceanography Centre, University of Southampton, Southampton, UK

Abstract

The integration of computational grids and data grids into a common problem solving environment enables collaboration between members of the GENIE*fy* project. In addition, state-of-the-art optimisation algorithms complement the component framework to provide a comprehensive toolset for Earth system modelling. In this paper, we present for the first time, the application of the non-dominated sorting genetic algorithm (NSGA-II) to perform a multiobjective tuning of a 3D atmosphere model. We then demonstrate how scripted database workflows enable the collective pooling of available resource at distributed client sites to collaboratively progress ensembles of simulations through to completion on the computational grid. A systematic study of the oceanic thermohaline circulation in a hierarchy of 3D atmosphere-ocean-sea-ice models is presented providing evidence for bi-stability in the Atlantic Meridional Overturning Circulation.

1 Introduction

The GENIE project (Grid ENabled Integrated Earth system model [1]) has created a Grid enabled component framework for the composition, execution and management of Earth System Models (ESMs). Mature simulation codes that model constituents of the Earth system (e.g. ocean, atmosphere, land surface, etc.) at varying resolution, complexity and dimensionality can be coupled together to provide a comprehensive hierarchy of climate models. The focus of the project is to create computationally efficient models for the study of the Earth system over millennial timescales and in particular to study ice age cycles and long-term human induced climate change.

Grid technology is a key enabler for the collaborative flexible coupling of constituent models, subsequent execution of the resulting ESMs and the management of the data that they generate. In this paper, we demonstrate how the flexible construction of ESMs in the GENIE framework in conjunction with the Grid software deployed for the project is exploited to tune a constituent component and then execute large ensemble simulations of computationally expensive models to study bi-stability in the oceanic "thermohaline circulation". We demonstrate how scripted database workflows enable the collective pooling of available resource at distributed client sites to progress an ensemble of simulations through to completion.

In this paper, we present the results of recent work extending the study of Marsh *et al* [2]. Section 2 presents the scientific problem that this work addresses. We discuss the Grid-enabled Problem Solving Environment (PSE) that we have exploited to perform these studies in Section 3. The results from a comprehensive study of bi-stability in a 3D atmosphere-ocean-sea-ice model are presented in Section 4. We discuss the merits of our approach in Section 5 and conclude in Section 6.

2 Scientific Challenge

A significant aspect of the climate system is the thermohaline circulation (THC), the name given to the system of large currents that connect and flow through the major oceans of the world. A principal component of the present day THC is the Gulf Stream in the North Atlantic which brings warm surface waters from the Gulf of Mexico to northern Europe. In the middle and high latitudes, heat and moisture from the ocean are lost to the atmosphere giving rise to the temperate climate in this region. As a consequence of this transfer the ocean waters become both cooler and more saline and hence increase in density. By a process of deep convection cool water sinks to the ocean floor and flows southwards until it reaches the

Southern Ocean. The north flowing surface current and the south bound deep current form a major part of the global ocean "Conveyor Belt" and are often referred to collectively as the Atlantic Meridional Overturning Circulation (MOC). The vast currents that circulate the globe because of the MOC are responsible for transport of heat energy and salinity and play a major role in determining the Earth's climate.

The strength of the Atlantic MOC is sensitive to the global hydrological cycle. In particular, variation in the atmospheric fresh water transport between the Atlantic and Pacific basins could lead to systematic changes in the THC. Since moisture transports by the atmosphere increase significantly under global warming scenarios, a key concern for climate change predictions is an understanding of how the THC may be affected under increased greenhouse gas forcing. It is possible that the system will react in a non-linear fashion to increased fresh water transports and the Atlantic MOC could collapse from its current "on" state into an "off" state, where no warm conveyor belt exists, and a colder, drier (more continental) winter climate may be expected in northern Europe. Studies using box models [3] and models of intermediate complexity (EMICs) such as GENIE-1 (C-GOLDSTEIN) [2], have found bi-stability in the properties of the THC; the "on" and "off" states can exist under virtually the same fresh water flux properties of the atmosphere depending on the initial conditions of the model (starting from an "on" or "off" state). However, the most comprehensive type of climate model, the coupled Atmosphere-Ocean General Circulation Models (AOGCMs), have yet to find any conclusive evidence for this bi-stability. In this paper, we extend the work performed with the GENIE-1 model (comprising a 3D ocean, 2D atmosphere and sea-ice) and present the first study of THC in the GENIE-2 model, a fully 3D ocean-atmosphere-sea-ice EMIC model from the GENIE framework. We thus take a step up the ladder of model complexity to study the behaviour of the same ocean model but now under a 3D dynamical atmosphere in place of the simple 2D energy moisture balance code (EMBM) of GENIE-1.

This study extends the work of Marsh [2] in two key areas. First, the new atmosphere model must be tuned to provide a reasonable climatology when coupled to the 3D ocean. We present, for the first time, the application of a multiobjective tuning algorithm to this problem. The second issue relates to computational complexity. Due to the need for shorter timesteps to handle atmospheric dynamics and the addition of a third (vertical) dimension in the atmosphere, the GENIE-2 model requires approximately two orders of magnitude more CPU time than GENIE-1 to simulate an equivalent time period. In Marsh [2] each GENIE-1 simulation required only a few hours of CPU time and the entire study was performed in about 3 days on a 200 node flocked Condor pool. However, for GENIE-2 models, individual simulations require ~5-10 days of continuous run time and a cycle stealing approach is no longer appropriate. Indeed, simply securing a sufficient number of CPUs over weekly timescales would be a significant challenge. In the remainder of this paper we show how the Grid computing software described in [4] is essential for the break down of large ensembles of lengthy simulations into manageable compute tasks. The GENIE data management system is used to mediate the execution of large ensemble studies enabling members of the project to pool their resource to perform these sub-tasks. Through a collaborative effort the more expensive study of GENIE-2 is enabled.

3 Collaborative PSE

GENIE has adopted software developed by the Geodise project [5] to build a distributed collaborative problem solving environment [4] for the study of new Earth system models. The Geodise Toolboxes integrate compute and data Grid functionality into the Matlab and Jython environments familiar to scientists and engineers. In particular, we have built upon the Geodise Database Toolbox to provide a shared Grid-enabled data management system, a central repository for output from GENIE Earth system models. An interface to the OPTIONS design search and optimisation package [6] is also exploited to provide the project with access to state-of-the-art optimisation methods which are used in model tuning.

3.1 Data Management System

The Geodise data model allows users to create descriptive metadata in Matlab / Jython data structures and associate that metadata with any file, variable or datagroup (logical aggregation of data) archived to the repository. The Geodise XML Toolbox is used to convert the metadata to a valid XML document which is then ingested and made accessible in the database. The GENIE database system augments this model by defining XML schemas that constrain the permissible metadata and improve

subsequent query performance. The interface to the database is exposed through web services and all access is secured and authenticated using X.509 certificates. Through a common client members of the project can access the database both programmatically and with a GUI interface.

Recent work on the Geodise database toolbox has further enhanced its functionality by supporting aggregated queries (e.g. max, count), datagroup sharing (i.e. users working on the same task can add data into a shared data group), and improved efficiency of data access control. These enhancements enable the scientist to compose more powerful and flexible scripted workflows which have the ability to discover, based on the content of the database, the status of an experiment and make further contributions to the study.

3.2 Task Farming

We exploit the GENIE database and the restart capabilities of the models to break down ensembles of long simulations into manageable compute tasks. By staging the restart files in the shared repository all members of the project can access the ensemble study, query against the metadata in the database to find new work units, submit that work to compute resources available to them and upload new results as they become available. Using the common client software project members can target local institutional clusters, Condor pools or remote resources available to them on the grid (Globus 2.4).

The task farming paradigm is ideally suited to Grid computing where multiple heterogeneous resources can be harnessed for the concurrent execution of work units. Examples of such systems are Nimrod/G [7] and GridSolve [8] which provide task farming capabilities in a grid environment. Nimrod/G allows users to specify a parametric study using a bespoke description language and upload the study to a central agent. The agent mediates the execution of the work on resources that are available to it. The GridSolve system works in a similar fashion, exploiting an agent to maintain details about available servers and then selecting resource on behalf of the user. In contrast we exploit the GENIE database system, providing scriptable logic on the client side to coordinate the task farming based on point-in-time information obtained from the central database. This allows more dynamic use of resource as the client is more easily installed within institutional boundaries. The study is mediated in persistent storage, as with Nimrod/G and GridSolve, allowing progress to be monitored and output data to be shared.

3.3 Collaborative study

The programmatic interface to the shared data repository allows the database to become an active part of an experiment workflow.

To define an ensemble in the database the coordinator of the study creates an experiment "datagroup", a logical entity in the database describing the purpose of the ensemble and acting as parent to a set of simulation definitions. The simulation entities record the details for the specific implementation of the model including the total number of timesteps to be performed and the output data to generate. This data structure within the database captures all of the information that is required for the study and its creation amounts to ~100 lines of configurable script for the scientist to edit.

Once a study has been defined the coordinator typically circulates the unique identifier to members of the project. If a user wishes to contribute compute resource for the progression of the experiment then this is the only piece of information that they require. They simply contribute to the study by invoking a "worker" script, specifying the experiment identifier and providing details of the amount of work they would like to submit to a particular resource. A user session is shown in Figure 1.

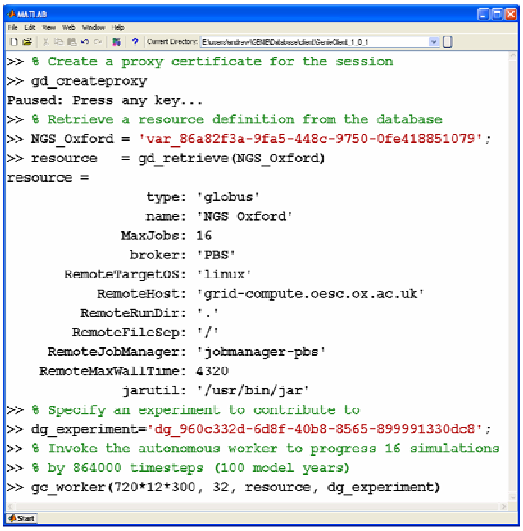


Figure 1: Typical user session contributing to an ensemble study.

The first action is to create a time-limited proxy certificate (to authenticate all actions on the Grid). The user then retrieves a resource definition from the database and invokes the autonomous "worker" script. The "worker"

allows a user to specify the number of timesteps to progress the members of the ensemble by and the number of compute jobs to submit to the specified resource.

The user specifies a resource by creating a data structure in Matlab capturing the information needed by the system to exploit that resource. We provide a utility function that prompts the user for the necessary information and then upload an appropriate data structure to the database. Once the resource definition is available in the database it is more common for the user to retrieve the definition and use it to submit work to the resource it describes. The worker script progresses through four stages of execution (see Figure 2):

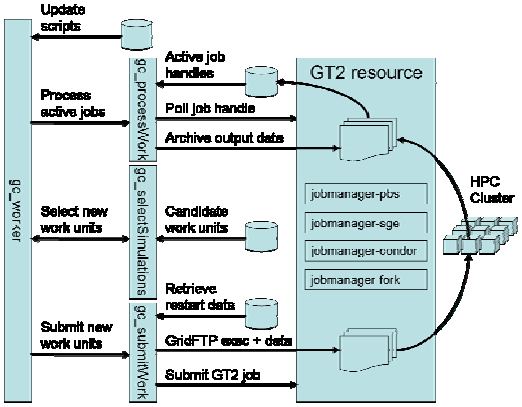


Figure 2: Scripted workflow of the autonomous "worker" interfacing to a resource managed by the Globus Toolkit (v2.4).

Stage 1: The worker interrogates the database for details of any changes to the scripts that define the experiment. Any new or updated scripts that have been added to the experiment are downloaded and installed in the user's local workspace. This provides the experiment coordinator the means to dynamically update the study and ensures that all participants are running the same study.

Stage 2: The worker invokes a post-processing script to ensure that any completed work units are archived to the database before new work units are requested. A list of active job handles is retrieved from the database and each handle is polled for its status. Any completed compute tasks are processed and the output from each is uploaded to the database. The files from which the model may be restarted are tagged as such. In the event of a model failure the post-processing will attempt to record the problem to allow future invocations of the worker to perform corrective action.

Stage 3: The worker invokes a selection script that queries the database for available compute tasks. Based on a point-in-time assessment of the progress of the study the system returns a list of work units that are available for progression. The list of jobs is prioritised so that the entire ensemble is progressed as evenly as possible; all inactive work units are returned first with those having least achieved timestep given highest priority.

Stage 4: The final action of the worker script is to submit the specified number of compute tasks to the resource that the user provided. Working through the list of compute tasks obtained in stage 3 the script retrieves the appropriate restart files from the database and transfers them, along with the model binary, to a unique directory on the remote compute resource. Once the model run is set up the script submits the job to the scheduler of the resource. The job handle returned by the scheduler is uploaded to the database. Subsequent invocations of a worker script on this experiment are then informed about all active compute tasks and can act accordingly.

It would be unreasonable to require a user to manually invoke the worker script on a regular basis and the most common mode of operation is to set up a scheduled task (Windows) or cron job (Linux) to automatically initiate the Matlab session. The user typically provides a script to run one or more workers that submit work to the resources available to the client. Through the regular invocation of the worker script the entire ensemble is progressed to completion without any additional input by the user. The progress of the study can be monitored at any point through a function call that queries the database for summary information about the experiment.

4 Results

We present the results of an end-to-end study of a small suite of GENIE-2 models. The first stage of the study performs a tuning of the IGCM atmosphere to provide a stable coupling to the GOLDSTEIN ocean component. Using an optimal parameter set we then define a number of ensembles in the database system for the IGCM coupled to the GOLDSTEIN model running on three computational meshes with varying horizontal resolution. Project members collaborate to progress the simulations through to completion on resource available to them both locally and nationally.

4.1 Multiobjective Tuning

An important aspect of Earth system model development is the tuning of free parameters so

that the simulated system produces a reasonable climatology. In Price [4] the OPTIONS design search and optimisation package [6] was used to apply a Genetic Algorithm to vary 30 free parameters in the IGCM and minimise a single objective measure of the mismatch between annually averaged model fields and observational data. This measure of model-data mismatch was reduced by ~36% and the resulting parameters are now the preferred set for further study using the IGCM. However, while this tuning produced a good improvement in the sea-surface temperature (SST), surface energy fluxes and horizontal wind stresses, there was little or no improvement in the precipitation and evaporation fields which comprised part of the tuning target. Analysis of the model state at this point (GAtuned) in parameter space also showed that seasonal properties of the model were not a good match for seasonally averaged observational data.

Recent studies have therefore adopted a multiobjective tuning strategy in order to provide targeted improvements in seasonally averaged fields of the model across physically significant groupings of state variables. In general, multiobjective optimisation methods seek to find a set of solutions in the design space that are superior to other points when all objectives are considered but may be inferior in a subset of those objectives. Such points lie on the Pareto front and are those for which no other point improves all objectives [9]. We have exploited an implementation of the non-dominated sorting genetic algorithm (NSGA-II) [10]. As described in [4] the GENIE model is exposed as a function that accepts as input the tuneable parameters and returns, after simulation, the multiple objective measures of model mismatch to observational data. The NSGA-II algorithm maintains a population of solutions like a GA but uses a different selection operator to create each successive generation. The algorithm ranks each member of the parent population according to its non-domination level and selects the members used to create the next population from the Pareto-optimal solutions that have maximum separation in objective space. The method seeks to reduce all objectives while maintaining a diverse set of solutions. The result of the optimisation is a set of points on the Pareto front from which a user can then apply domain knowledge to select the best candidates.

The IGCM atmosphere component was tuned using the NSGA-II algorithm applied over 50 generations using a population size of 100. 32 free parameters in the genie_ig_sl_sl model (IGCM atmosphere, slab ocean, slab sea-ice) were varied and 2 constraints were applied. Three objective functions were defined to provide improvements in the seasonal averages of the surface energy fluxes (OBJ1), the precipitation and evaporation properties (OBJ2) and the surface wind stresses (OBJ3). The model runs were performed on a ~1400 node Condor pool at the University of Southampton with each generation taking approximately three hours to complete. The algorithm generated a Pareto optimal set of solutions comprising 117 members. The results of the tuning exercise are summarised in Table 1 comparing the 'best' result (arbitrarily selected as the point with minimum sum of the three objectives) of the Pareto study with the objectives evaluated at the default and GAtuned points in parameter space.

We first point out that the precipitation and evaporation in the model (OBJ2) saw little improvement in the original GAtuned study over the default parameters even though these fields were part of that tuning study. The GApareto result has produced significant improvements in all three objective functions as desired and provided a better match to the observational data. The scale of these improvements is also greater then the original GAtuned study, but we note that this is primarily due to the seasonal rather than annual averages that have been used to comprise the tuning target. That is to say that we are comparing the seasonal measures of fitness at a point that was obtained when optimising annual averages – we should expect a greater improvement in the seasonal study. The improvements in the precipitation and evaporation fields are not as great as the improvements made in the other two objectives and we note that there are probably fundamental limits to the model's ability to simulate precipitation and evaporation that tuning alone cannot overcome.

Name	Default	GAtuned	GApareto
OBJ1	4.47	3.68	3.23
OBJ2	3.87	3.84	3.41
OBJ3	3.11	2.32	2.08

Table 1: Values of the three objective functions evaluated at the default, GAtuned and GApareto points in parameter space.

4.2 Study of THC bi-stability

We have performed a number of ensemble studies of the GENIE-2 ig-go-sl model consisting of the 3D IGCM atmosphere code coupled to the 3D GOLDSTEIN ocean model

with a slab sea-ice component. This model typically requires approximately 5 days of continuous run time to perform a 1000 year simulation of the Earth system, which is long enough for the global THC to approach equilibrium. In the Atlantic Ocean, the THC is characterised by a "meridional overturning", which is diagnosed in the model as a streamfunction in the vertical-meridional plane. The units of this meridional overturning streamfunction are 10^6 m^3s^{-1}, or Sverdrups (Sv). The sign convention is positive for a clockwise circulation viewed from the east, corresponding to surface northward flow, northern sinking and southward deep flow. This circulation pattern corresponds to the Conveyor Belt mode of the THC, the strength of which is given by the maximum of the overturning streamfunction. In order to obtain a realistic Conveyor Belt in GENIE-2 (as in other similar models), it is necessary to include a degree of surface freshwater flux correction to ensure a sufficient Atlantic-Pacific surface salinity contrast. We have studied the Atlantic maximum overturning rate under a range of this freshwater flux correction in a three-level hierarchy of GENIE-2, across which we vary the resolution of the horizontal mesh in the ocean component.

Twelve ensemble studies were executed by three members of the project using database client installations at their local institutions. The unique identifier for each study was circulated amongst members of the project willing to contribute resource and regular invocations of the client "worker" script were scheduled at each site to submit model runs to a number of computational resources around the UK. These resources included five nodes of the UK National Grid Service (http://www.ngs.ac.uk), three Beowulf clusters available at the Universities of East Anglia and Southampton and a large Condor pool (>1,400 nodes) at the University of Southampton.

The ensemble studies of the GENIE-2 model presented here were carried out during a 14 week period from December 2005 until March 2006. In total five client installations were used to perform:

- 319 GENIE-2 simulations
- 3,736 compute tasks
- 46,992 CPU hrs (some timings estimated)
- 362,000 GENIE-2 model years

The daily breakdown of resource usage is plotted in Figure 3. The rate of progression of the studies was largely determined by the amount of available work in the system at any given time. The overheads in performing queries on the database and the upload and retrieval of model data files did not adversely impact the performance of the system.

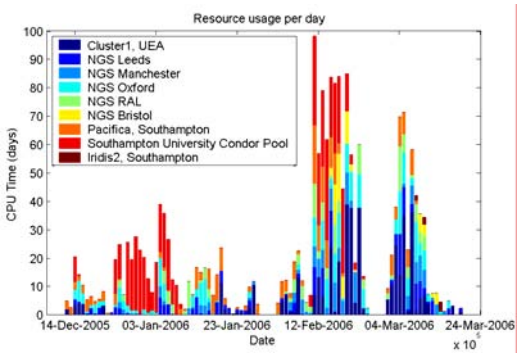


Figure 3: Resource usage for the twelve ensemble studies.

The breakdown of client contributions is presented in Figure 4a. The distribution of submitted jobs roughly reflects the amount of resource available to the client in each case. E.g. the client responsible for ~50% of the work was the only submission node on the large condor pool and was also used to submit jobs to the National Grid Service. The distribution of jobs across the computational resources is presented in Figure 4b and illustrates that work was distributed evenly to the available resources. By mediating the study through a shared central repository the coordinating scientist has had the work performed over a collected pool of resource, much of which is behind institutional firewalls and probably not directly accessible to him/her. The system also enables us to introduce new resource as a study is performed and target the most appropriate platforms.

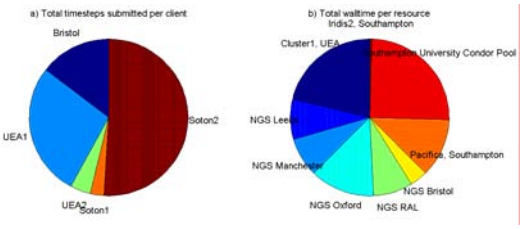


Figure 4: Distributions of client submissions and resource usage.

The data from these runs is being processed and a more detailed analysis of the results of these studies will be the subject of a separate paper [11]. We briefly present the initial findings of this experiment.

The effect of varying a fresh water flux correction parameter in the atmosphere component (IGCM) of a suite of GENIE-2 models with different ocean grid resolutions has been studied. Six of the ensembles are plotted in

Figure 5 showing the strength of the maximum Atlantic overturning circulation for ensemble members across three meshes (36x36 equal area, 72x72 equal area, 64x32 lat-lon) initialised from states with default present day circulation (r1) and collapsed circulation (r0). Maximum overturning rates are obtained as averages over the last 100 years of the 1000-year simulations.

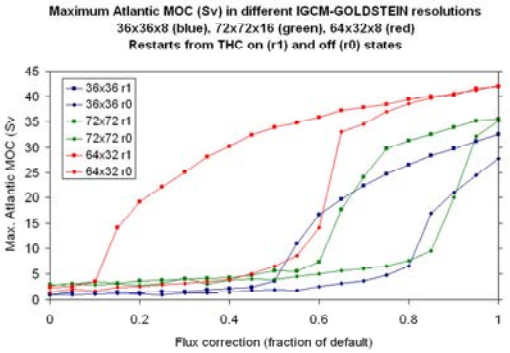


Figure 5: Maximum Atlantic overturning circulation (MOC) across six ensembles, for three different horizontal meshes of the ocean component.

Bi-stability in the Atlantic MOC is clearly evident in all three model variants. Under the same atmospheric conditions there is a range of fresh water corrections for which the ocean THC is stable in both the "on" and "off" states depending on initial conditions. The locations and widths of the hysteresis loops as a function of ocean mesh resolution are also an important feature of these results. The increase of grid resolution from 36x36 to 72x72 (effectively multiplying by volume the number of grid cells by 8) shows a slight narrowing of the hysteresis loop and steeper transitions. The use of an ocean grid at the same resolution as the atmosphere (64x32) exhibits a much wider loop and is positioned at a different point in the parameter space. The differences in these results are likely attributable to two factors; a) the interpolation employed in coupling the ocean and atmosphere models and b) the different resolutions of the ocean grids at the mid-latitudes which mean important processes are better resolved at 64x32. In the 64x32 model, there are more grid points in the east-west direction giving better resolution of zonal pressure gradients in the ocean, better THC structure, and improved northward transport of high salinity water by the THC itself, reducing the need for surface freshwater flux correction. These results tentatively support the notion that more complex models can exhibit bi-stability in the states of the THC.

5 Discussion and Future Work

Programmatic access to a shared central database provides the means to mediate the collaborative study of Earth System models in the GENIE project. In contrast to other grid-enabled task farming systems, such as Nimrod/G and GridSolve, we exploit client side scripted database workflows in place of a central agent to mediate the execution of our studies. Our system allows resource to be dynamically introduced to studies, but we achieve this at the expense of execution control. Systems with a central agent can build execution plans, target jobs at the most appropriate platform and provide guarantees for completion times of experiments. However, administrators of the agent system must ensure that this central server can communicate with all resource to be targeted. If available resource has not been exposed to the grid then there is no way of using it. By providing a rich client within the institutional boundary we maximise our ability to exploit resource. However, our system is passive and relies upon the users to contribute their resource and establish regular invocations of their client system. While this provides a scalable and robust solution it cannot provide any guarantees about completion time. A publish / subscribe approach would overcome this shortcoming but would require the development of an active message queue in the database system. We will investigate this possibility. The overheads in moving data through a client are avoided in the task farming systems but the dynamic allocation of compute tasks through multiple clients offsets this issue.

The definition and upload of a model study involves some simple edits to a template script to specify the unique experiment definition in the database. Once created the experiment coordinator circulates the unique identifier for the study and members of the project can then choose whether to contribute their resource. The effort involved in contributing to a study is minimal because the "worker" script functions autonomously. A user typically schedules a cron job or windows scheduled task to execute several times a day and then has no further interaction with the system. The scheduled task initiates the client and invokes the "worker" using the experiment identifier. As long as a valid proxy certificate exists, the system attempts to contribute work until the study is complete. A project member may stop their scheduled task at any time and withdraw their resource, introduce new resource, and drop in and out of the experiment as they see fit.

As a project we are able to exploit resources across institutional boundaries without any modification to firewalls. As long as a client can talk to the database and the GridFTP server on standard ports then we can exploit a user's local rights to resource at their institution. Our system pulls the work to a client within the institutional boundary rather than needing to push it through.

The database provides a very robust and fault tolerant environment to perform our work. The entire study is held in persistent storage and progress is only achieved through the successful update of the database contents. Problems encountered on the grid, external to the database, do not have any direct impact on a study. Failed work units are automatically re-submitted by later invocations of the "worker" script. The system also provides a means to maximize our responsible use of available resource. Since the database maintains a record of the number of tasks running on each server the client can respect specified usage limits. Once each resource reaches its limit the client will move on to the next and the system therefore keeps available compute power busy.

6 Conclusions

Members of the GENIE project collaborated to perform large-scale ensemble calculations on hierarchies of Earth System Model from the GENIE framework. Using a common Grid-enabled client interfacing to the shared GENIE database, users pooled their resource to contribute to ensemble studies defined in the database. Armed with the unique system identifier for a particular study a user simply invoked an autonomous "worker" script specifying the amount of work they would like to contribute on a given resource. Through regular invocations of the client by multiple members of the project the ensembles were progressed through to completion. The system provides a robust and fault tolerant means to carry out large ensembles of lengthy model runs as progress is only achieved by the successful upload of data to the database. Previous studies of THC stability in models have been restricted to inter-comparison of disparate models of generally lower complexity than GENIE-2 [12]. Here e-Science technology has allowed us to perform the first systematic investigation of THC stability within a single model hierarchy.

Acknowledgements

The GENIE and GENIE*fy* projects are funded by the Natural Environment Research Council (NER/T/S/2002/00217 & NE/C515904) through the e-Science programme. The authors would like to acknowledge the use of the UK National Grid Service in carrying out this work.

References

[1] The GENIE project. http://www.genie.ac.uk

[2] Marsh R., A. Yool, T. M. Lenton, M. Y. Gulamali, N. R. Edwards, J. G. Shepherd, M. Krznaric, S. Newhouse and S. J. Cox, 2004. Bistability of the thermohaline circulation identified through comprehensive 2-parameter sweeps of an efficient climate model. Climate Dynamics, **23**(7–8), 761-777.

[3] Wang, H. and G. E. Birchfield, 1992. An energy-salinity balance climate model: Water vapor transport as a cause of changes in the global thermohaline circulation, J. Geophys. Res., **97**, 2335-2346.

[4] Price, A. R., Xue, G., Yool, A., Lunt, D. J., Valdes, P. J., Lenton, T. M., Wason, J. L., Pound, G. E., Cox, S. J. and the GENIE team, 2006. Optimisation of integrated Earth System Model components using Grid-enabled data management and computation. Concurrency and Computation: Practice and Experience, DOI: 10.1002/cpe.1046.

[5] The Geodise project. http://www.geodise.org

[6] Keane, A. J., 2003. The OPTIONS design exploration system: reference manual and user guide. http://www.soton.ac.uk/~ajk/options.ps

[7] Abramson, D., Buyya, R. and Giddy, J., 2002. A computational economy for grid computing and its implementation in the Nimrod-G resource broker, Future Generation Computer Systems, **18**(8), 1061-1074.

[8] YarKhan, A., Seymour, K., Sagi, K., Shi, Z. and Dongarra, J., 2006. Recent Developments in Gridsolve, International Journal of High Performance Computing Applications, **20**(1), 131-141. DOI: 10.1177/1094342006061893

[9] Srinivas, N. and Deb, K., 1995. Multiobjective function optimization using nondominated sorting genetic algorithms, Evol. Comp. J., **2**(3), 221-248.

[10]Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, **6**(2), 182 – 197.

[11]Lenton, T. M., *et al*., 2006. A modular, scalable, Grid ENabled Integrated Earth system modelling framework: Effect of dynamical atmosphere model, surface grid, and ocean resolution on the stability of the thermohaline circulation. Clim. Dyn. submitted.

[12]Rahmstorf, S., Crucifix, M., Ganopolski, A., Goosse, H., Kamenkovich, I., Knutti, R., Lohmann, G., Marsh, R., Mysak, L. A., Wang, Z., and A. Weaver, 2005. Thermohaline circulation hysteresis: a model intercomparison. *Geophys. Res. Lett.*, **32**, L23605 10.1029/2005GL023655.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

GridPP: From Prototype to Production

D.I. Britton (Imperial College), S. Burke (CCLRC), A.J. Cass (CERN),
P.E.L. Clarke (University of Edinburgh), J.C. Coles (CCLRC), A.T. Doyle (University of Glasgow),
N.I. Geddes (CCLRC), J.C. Gordon (CCLRC), R.W.L. Jones (Lancaster University),
D.P. Kelsey (CCLRC), S.L. Lloyd (QMUL), R.P. Middleton (CCLRC),
D. Newbold (University of Bristol), S.E. Pearce (QMUL)
on behalf of the GridPP Collaboration.

Abstract

GridPP is a £33m, 6-year project funded by PPARC that aims to establish a Grid for UK Particle Physics in time for the turn on of the CERN Large Hadron Collider (LHC) in 2007. Over the last three years, a prototype Grid has been developed and put into production with computational resources that have increased by a factor of 100. GridPP is now about halfway through its second phase, the move from prototype to production is well underway though many challenges remain.

1. The LHC Computing Challenge

After more than a decade of work, the world's highest energy particle accelerator, the Large Hadron Collider (LHC), and the associated detectors come on line in 2007 at CERN in Geneva. With a design luminosity of 800,000,000 proton-proton interactions per second, the 100,000,000 electronic channels embedded in each of the four detectors will produce around 10 Petabytes of data per year. Buried in that landslide of data, perhaps at the level of 1 part in 10^{13}, physicists hope to find the rare signature of a Higgs particle. Discovering the nature of the Higgs sector (one or more physically observable particles) allows the origins of mass in the universe to be established.

A close relationship has existed between particle physics and computing for the last quarter of a century. Driven by economic, political, and performance issues Particle Physicists have moved from the gold-standard of service and performance provided by main-frames, through smaller institutional based single machines, to modest sized clusters based solutions. The Grid, a global and heterogeneous aggregation of hardware clusters, is the latest step along this path, which strives to minimise the computing cost by the use of commodity hardware; provide scalability to a size beyond that of mainframes; and deliver a quality of service sufficient for the task primarily by relying on redundancy and fault-tolerance to balance the intrinsic unreliability of individual components. The Grid model matches the globally diverse nature of the Particle Physics experiment collaborations, providing politically and financially acceptable solutions to an otherwise intractable computing problem.

The data from the LHC detectors are filtered in quasi-real time by dedicated online trigger hardware and software algorithms. The selected raw data will stream from the detectors to the Tier-0 computing centre at CERN and individual trigger streams will also be channelled to specific members of a global network of a dozen Tier-1 centres. The raw data are reconstructed and calibrated in a CPU intensive process, before being catalogued and archived as Event Summary Datasets (ESD). The data are further refined and rarefied to produce Analysis Object Datasets (AOD) and Tagged samples. All these datasets may be used subsequently for data analysis and metadata about the data is also required to be compiled and catalogued. The raw data are complimented by a comparable quantity of simulated data that are generated predominantly at smaller regional Tier-2 sites before being processed in a similar manner to the raw data in order to understand detector performance, calibration, backgrounds, and analysis techniques. The computing requirements are enormous: In 2008, the first full year of data taking, CPU capacity of 140 million SPECint2000 (140,000 3GHz processors), 60 PB of disk storage and 50 PB of mass storage will be needed globally. The hierarchy of Tier centres represents an optimisation of the resources mapped to the functionality and level of service required for different parts of this problem. On the one hand this recognises that there are economise of scale to be gained in the management and operations of computing resources, particularly commodity hardware where there is only basic level vendor support; on the other hand it acknowledges that

not all parts of the problem need to the same services or quality of service and that substantial benefits in cost and scale can also be gained by embracing an architecture where institutes, regions, or even Countries, can plug-and-play. This, then, is the optimisation afforded by the Grid approach.

2. Overview of GridPP

Since September 2001, GridPP has striven to develop and deploy a highly functional Grid across the UK as part of the LHC Computing Grid (LCG)[1]. Working with European EDG and latterly EGEE projects [2], GridPP helped develop middleware adopted by LCG. This, together with contributions from the US-based Globus [3] and Condor [4] projects, has formed the LCG releases which have been deployed throughout the UK on a Grid consisting presently of more than 4000 CPUs and 0.65 PB of storage. The UK HEP Grid is anchored by the Tier-1[5] centre at the Rutherford Appleton Laboratory (RAL) and four distributed Tier-2 [6] centres known as ScotGrid, NorthGrid, SouthGrid and the London Tier-2. There are 16 UK sites which form an integral part of the joint LHC/EGEE computing Grid with 40,000 CPUs and access to 10 PB of storage, stretching from the Far-East to North America.

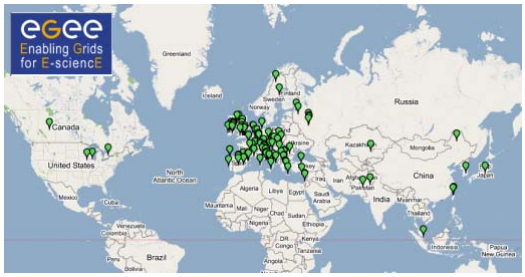


Figure 1: The Global EGEE/LCG Grid

3. Performance Review

The current phase of GridPP moves the UK HEP Grid from a prototype to a production platform. Whilst progress can be monitored by milestones and metrics, success can ultimately only be established by the widespread and successful use of substantial resources by the community. Collecting information about Grid use is, in itself, a Grid challenge. GridPP sites form the majority of the EGEE UK and Ireland region (UKI), with RAL as the Regional Operations Centre (ROC). RAL also runs the Grid Operations Centre (GOC) [8] which maintains a database of information about all sites and provides a number of monitoring and accounting tools that provide insight and information. At the basic level, the Site Functional Test (SFT), a small test job that runs many times a day at each site, determines the availability of the main Grid functions. Similarly, the Grid Status Monitor (GStat) retrieves information published by each site about its status. Figure-3 shows the average CPU availability by region for April 2006 derived from sites passing or failing the SFT. Although this particular data set is not complete (and an improved metric is being released in July) it can be seen that within Europe the UKI region (second entry from the left) made a significant contribution with 90% of the total of just of 4000 CPUs being available on average.

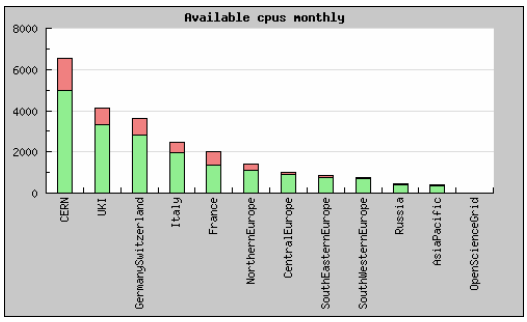


Figure-3: Average CPU availability for April 2006. CPUs at a site are deemed available (Green) when the site passes the SFT or unavailable (Red) if the site fails.

In addition to the GOC database containing Grid-wide information, statistics are also recorded at the RAL Tier-1 centre using Ganglia to monitor CPU load, memory usage and queue data from the batch system. Figure-4 shows the usage by Virtual Organisation (VO) for 2005. Full capacity is roughly the top of the graph so the Tier-1 facility was running around 90% of capacity for the latter half of the year, though about half of this was non-Grid use, predominantly by the b-factory.

In order to understand the apparently low CPU utilisation in the first half of 2005, a detailed analysis of batch job efficiency was carried out where the efficiency is the ratio of CPU time to elapsed time. A highly CPU intensive batch job can achieve 95-98% utilisation of a CPU, an I/O intensive job is more likely to be around 85-95% utilisation of a CPU, and jobs waiting for busy resources can vary from 0-100% efficient. As can be seen from Figure-5, the overall efficiency was rather low during the second quarter of 2005 until the applications and their data-access patterns were better understood. When CPU time is corrected by job efficiency (to give job elapsed time), it is apparent (Figure-

6) that the farm ran with greater than 70% occupancy for most of the year, rising to 100% in December.

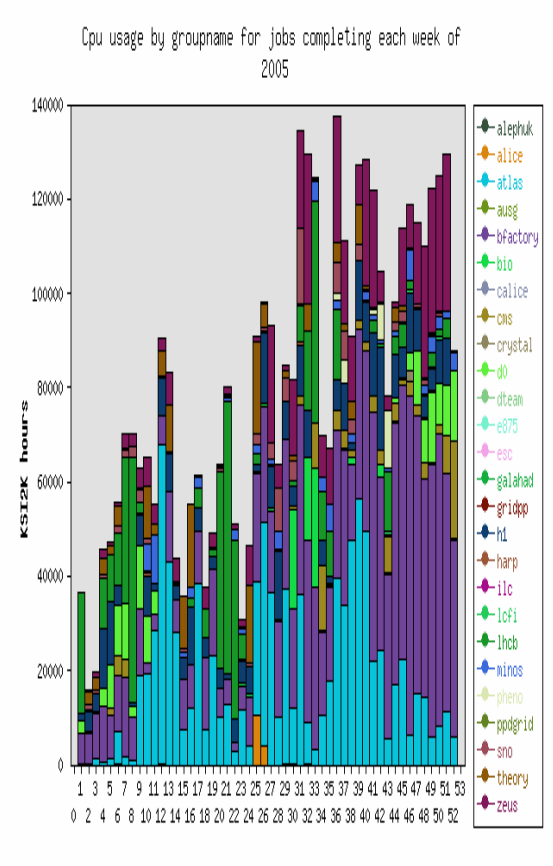


Figure-4: Tier-1 CPU use for 2005.

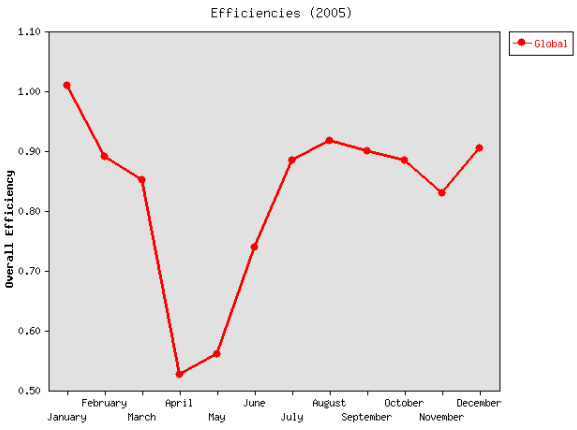


Figure-5: CPU efficiency (CPU/Wall -time).

The efficiency has continued to improve in 2006 with many experiments maintaining efficiencies well over 90%. The 2006 efficiency by Virtual Organisation is showing in Figure-7 below ("DTEAM" refers to the development team and debugging leads to low observed efficiency).

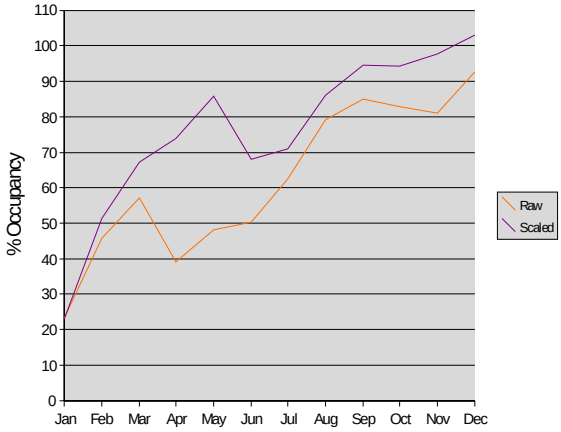


Figure-6 Tier-1 Calculated occupancy (purple curve is scaled for observed efficiency).

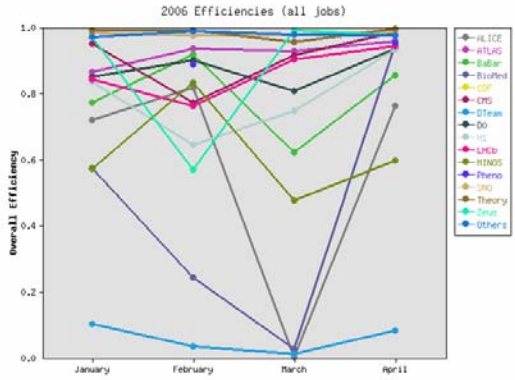


Figure-7: Job Efficiency (CPU-time/Wall-time) for 2006, by Virtual Organisation.

The hardware deployed and managed through GridPP is guided by a Memorandum of Understanding (MOU) signed by PPARC with CERN which defines the intended provision of hardware for the LHC experiments. In addition, GridPP has internal MOUs with the Tier-2 institutes which outline the hardware provision intended. However, actual purchases are optimised to reflect the anticipated needs during the near-term future so that, overall, the hardware resources can be maximised. In 2005 the Tier-1 hardware purchase was delayed and the hardware at many of the Tier-2 sites, particularly disk, ramped up more slowly than originally planned. Tables 1 and 2 at the end of this paper show the CPU and Storage installed at the Tier-1 and Tier-2s over the last five quarters, compared with the original plans contained in the MOUs. The Tier-1 has provisioned 60% of the CPU and 70% of the storage (which includes Tape) originally envisaged. The Tier-2s have been somewhat slower to ramp-up and although significant CPU

was added at the start of 2006 taking the overall provision to 75%, the storage is still much lower than planned at 34%. All these numbers need to be understood in the perspective of the actual usage, contained in Tables 3 and 4.

The usage tables show the resources allocated to the LCG Grid, which means it was declared via the LCG/EGEE mechanisms and monitored via the Site Functional Tests, with storage via an SRM (Storage Resource Manager- a protocol for managing storage) interface. The tables also show the fraction of this allocation that was used or, more precisely, the fraction of use that was recorded by the Grid Operations Centre for CPU and the GridPP SRM Storage accounting for disk and tape. There are a number of caveats associated with the accounting system; most notably that it currently does not account usage at Cambridge and London LeSC due to their use of Condor and Sun Grid Engine respectively. A preliminary new version of APEL (Accounting Processor for Event Logs: an accounting system which parses log files to extract and then publish job information) with Condor support has now been released to Cambridge for testing. Nevertheless, despite these known inefficiencies in the accounting, it is apparent that there was little pressure on the Tier-2 resources in 2005. Part of this is explained by a lack of confidence in the quality of Tier-2 storage by the experiments and GridPP is working to build explicit relationships between individual sites and experiments in order to create a better understanding of needs and services.

The Tier-1 is considered by GridPP to have delivered for all the experiments at the required target levels in 2005. Overall the UK Tier-1 centre delivered 29% of the CPU of the LCG Tier-0 and Tier-1 centres. The Tier-2s are also considered to have delivered more than the required capacity. Currently, the Tier-2s account for twice the delivered CPU and 1.5x the storage at the Tier-1. One of the challenges for 2006 is to achieve a more precise view of Grid usage and to increase the usage fraction of Tier-2 resources.

Whilst many of the statistics above address the scale of the Grid, reliability is also critical. The EGEE-JRA2 project [9] addresses the issue of Quality Assurance and the Figure-8 shows the success rate of jobs run in the UK for the period May-05 to May-06. Information by Virtual Organisation, and average wait times, are also available from the JRA2 web-site.

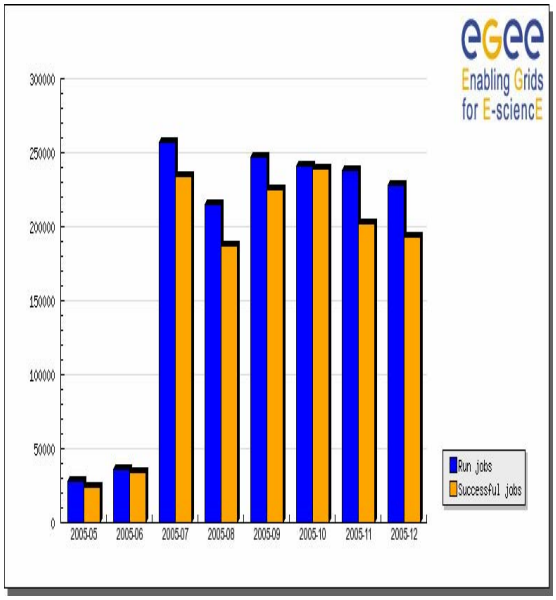


Figure-8: Job efficiency in the UK:

4. Service Challenges

The LCG has planned, and is executing, a series of world-wide Services Challenges designed to stress test the infrastructure and establish, incrementally, the levels of service and throughput needed for the LHC computing challenge. In the autumn of 2005, and through into the New Year, GridPP participated in Service Challenge-3. One aspect of this challenge was to establish Tier-0 to Tier-1 transfer rates of 150 Mbytes/sec (Disk to Disk) and 50 Mbytes/sec (Disk to Tape). Although the initial tests in July only achieved about half these targets (and with poor stability) by January 2006 the target rates had been established. Figure-9 shows a snapshot of the Tier-0 to Tier-1 disk-disk transfers for all Tier-1s in January 2006.

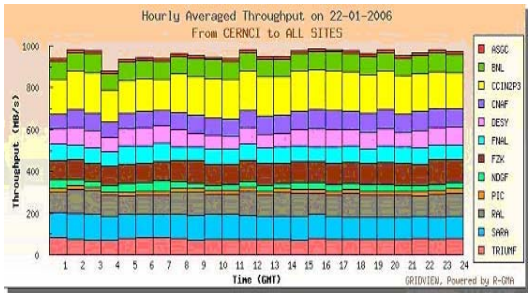


Figure-9: SC3 Tier-0 to Tier-1 Throughput.

Currently GridPP are engaged in the LCG Service Challenge-4 with goals that include ramping up the Tier-0 to Tier-1 transfer rates to full nominal rates (to tape); to identify and validate all other production data flows (Tier-x

to Tier-y); to increase Tier-2 participation from 20 sites worldwide in April 2006 to 40 by September; to broaden focus from production to analysis (where there are many more users); and to streamline Operations & User Support building on existing efforts. At the time of writing, the UK Tier-1 was sustaining disk-disk transfer rates to CERN of up to 160 Mbytes/sec and Tape-Disk rates of 40 Mbytes/sec. Figure-10 shows a snapshot of disk-disk though-put for a week in April 2006 and it can be seen that that the total concurrent flow from the Tier-0 is close to the target of 1600 Mbytes/Sec.

Figure-10: SC4 Tier-0 to Tier-1 disk to disk transfers.

In addition GridPP is conducting transfer tests between the RAL Tier-1 and each of the UK Tier-2 sites using the File Transfer Service (FTS) developed as part of the gLite [10] middleware stack. The target rate is a minimum of 250 Mbytes/sec for just reading or writing and 200 Mbytes/sec for simultaneous read and write. The eventual aim is to demonstrate that this can be sustained over long periods. Initial tests of separate read and writes have now been completed with 11 of the 19 sites exceeding the targets in at least one direction and 7 exceeding them in both. The highest speeds obtained were over a lightpath from Lancaster to RAL where a >900 Mbits/sec transfer rate was sustained for more than 90 hours and 1Gbit/sec was exceeded for significant periods (Figure-11).

5. Middleware Developments

GridPP contributes to middleware development in a number of areas, mainly through the EGEE project. The workload management group at Imperial have integrated the Sun Grid Engine (SGE) with the EGEE middleware, creating virtual batch queues to overcome the lack of traditional batch queues within SGE. An interface to the APEL accounting system has also been provided and is being tested. Following the decision to install CASTOR2 at RAL to replace existing ADS tape service, in tandem with the installation of the new Tape Robot, the storage group has agreed to contribute the SRM2 storage manager interface

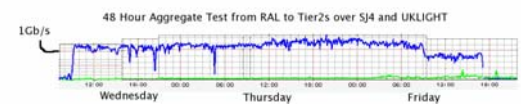


Figure-11: SC4 Tier-1 to Tier-2 transfer.

for Castor. The development of the R-GMA monitoring system, an implementation of the Grid Monitoring Architecture (GMA) has continued. GMA models the information and monitoring system infrastructure of a grid as a set of consumers (which request information), producers (which provide information) and a registry, which mediates the communication between the producers and consumers. The recent developments have greatly improving the robustness of the system and bringing significant improvements to the stability of the code deployed by EGEE/LCG on the production system. The credit to R-GMA may be noted on the bottom right-hand corners of Figures 9 and 10 in this paper and R-GMA is also currently used with the CMS "Dashboard". A major re-factored release of R-GMA was made for gLite-1.5. Similarly, GridSite [11] was updated for inclusion in gLite-1.5 where it provides containerised services for hosting VO boxes (machines specific to individual virtual organisations that run VO-specific services such as data management: An approach which, in principle, is a security concern) and support for hybrid HTTPS/HTTP file transfers (referred to as "GridHTTP") to the htcp tool used by EGEE. GridSiteWiki [12] has been developed, which allows the use of a Grid Certificate access to a WIKI, preventing unauthorised access, and which is in regular used by GridPP.

6. RealTime Monitor

The RealTime Monitor [13], developed as a spin-off from the GridPP portal work, displays all the jobs submitted through the resource brokers, monitored through a mySQL database connection. Since the information is continually updated, the job movement around the world map is seen in real-time. Figure-12 shows a screen-shot from the monitor. The green dots show running jobs and the lines show the movement of jobs to or from a resource broker. Not shown, but available from other tabs, are Phedex transfers (CMS proprietary data transfers) and the current usage of various switch light path links.

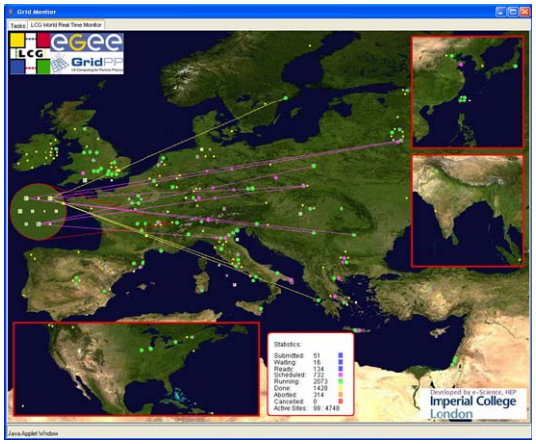


Figure-12: The RealTime Monitor

7. The Future

From a pragmatic point of view, we have come a long way down the Grid road, perhaps even far enough to recognise how far we still have to go. On the one hand, we now have a working Grid with resources that have grown by a factor of 100 over the last three years. On the other hand, in the next three years the Grid has to increase in scale by another factor of 10 and make large strides in functionality, robustness, and usability. In particular, the current Grid is largely used for fairly coordinated use by a relatively small number of active users (Figure-13 shows the number of users from the four LHC experiments combined during 2005).

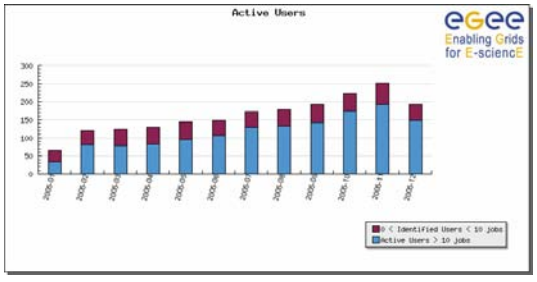


Figure 13: LHC experiment Grid users.

The future Grid must provide a platform, not only for coordinated production and reconstruction, but also for much more responsive (sometime called "chaotic") use by a much larger community intent on individual analyses. The LHCb collaboration has taken some initial steps and their DIRAC workload management system enables these shorter analysis jobs to be submitted with higher internal priority compared to background production jobs.

Some of the required functionality that is currently missing includes the ability to chain jobs through the Resource Broker and the ability to pin data for subsequent access. Without these two facilities, the Grid will become grossly inefficient as jobs occupy CPU waiting for data to be staged. The whole area of data movement and data management is underdeveloped at the Grid level and individual experiments have currently resorted to proprietary solutions which have spawned the need for experiment-specific persistent services (so called VO-boxes) at individual sites, which introduce security and scalability concerns. Although these are now envisaged to be limited to Tier-0 and Tier-1 sites, the better way forward would be to incorporate common services in the upper-level middleware of the Grid and experiment-specific services within the experiment software itself. Similarly, generic metadata handling remains a challenge and is a potential area of concern as the move is made to live analysis where perhaps hundreds of users are simultaneously making repeated and complex relational queries on databases that grow at (in the case of TAG data) by about 4TB a year. Finally, debugging on the Grid is notoriously difficult even for experts and reproducing the conditions of a failure, logging and error reporting all need to be raised to the level expected from PBS/LSF/NQS.

In the next 18 months, GridPP needs to successfully complete the GridPP2 work programme defined in the Project Map; deploy the new middleware releases based on gLite3; continue to participate in Service Challenge 4 and moving into the service phase; the hardware promised in the Memorandum of Understanding with CERN, now signed by PPARC, must be provided; the user communities must be developed and served; and a proposal for a 7 month extension of GridPP2 and a subsequent three year GridPP3 project must be developed.

References

[1] http://lcg.web.cern.ch/LCG/
[2] http://www.eu-egee.org/
[3] http://www.globus.org/
[4] http://www.cs.wisc.edu/condor/
[5] http://www.gridpp.ac.uk/tier1a/
[6] http://www.gridpp.ac.uk/tier2/
[7] http://www.gridpp.ac.uk/pmb/ProjectManagement/GridPP2_ProjectMap_6.htm
[8] http://goc.grid-support.ac.uk/gridsite/gocmain/
[9] http://egee-jra2.web.cern.ch/EGEE-JRA2/
[10] http://www.glite.org/
[11] http://www.gridsite.org/
[12] http://wiki.gridpp.ac.uk/wiki/GridPPwiki:About
[13] http://gridportal.hep.ph.ic.ac.uk/rtm

CPU Capacity [KSI2K]							
	Delivered (i.e. Installed)					Promised	Current
	2005-Q1	2005-Q2	2005-Q3	2005-Q4	2006-Q1	MOU 2005	Ratio
London	910	910	935	1049	1049	1351	78%
ScotGrid	32	186	237	273	354	340	104%
SouthGrid	354	483	492	508	516	667	77%
NorthGrid	205	750	750	776	1783	2602	69%
Total Tier-2	**1502**	**2329**	**2413**	**2607**	**3703**	**4960**	**75%**
RAL Tier-1	**830**	**830**	**830**	**830**	**830**	**1282**	**65%**

Table-1: Delivered CPU

Storage Capacity [TB]							
	Delivered (i.e. Installed)					Promised	Current
	2005-Q1	2005-Q2	2005-Q3	2005-Q4	2006-Q1	MOU 2005	Ratio
London	32	13	19	37	38	102	37%
ScotGrid	9	23	36	45	45	90	49%
SouthGrid	39	40	40	48	48	46	105%
NorthGrid	14	48	84	86	132	543	24%
Total Tier-2	**93**	**124**	**178**	**216**	**263**	**781**	**34%**
RAL Tier-1	**180**	**325**	**410**	**440**	**440**	**629**	**70%**

Table-2: Delivered Storage

CPU Usage [KSI2K]										
	Available (i.e. Allocated to LCG)					Fraction of Allocation Used (accounted)				
	2005-Q1	2005-Q2	2005-Q3	2005-Q4	2006-Q1	2005-Q1	2005-Q2	2005-Q3	2005-Q4	2006-Q1
London	557	542	592	884	884	3.2%	13.8%	39.2%	10.3%	22.7%
ScotGrid	31	78	237	237	182	0.1%	0.8%	4.9%	6.4%	31.9%
SouthGrid	151	205	207	243	243	4.4%	18.9%	47.3%	31.0%	45.3%
NorthGrid	311	745	765	772	1777	1.1%	9.2%	10.6%	10.1%	18.6%
Total Tier-2	**1050**	**1569**	**1800**	**2136**	**3086**	**2.6%**	**11.5%**	**22.9%**	**12.2%**	**22.7%**
RAL Tier-1	**444**	**444**	**444**	**444**	**444**	**49.8%**	**69.8%**	**67.8%**	**26.3%**	**77.0%**

Table-3: LCG CPU Usage

Disk Usage [KSI2K]										
	Available (i.e. Allocated to LCG)					Fraction of Allocation Used (accounted)				
	2005-Q1	2005-Q2	2005-Q3	2005-Q4	2006-Q1	2005-Q1	2005-Q2	2005-Q3	2005-Q4	2006-Q1
London	1.3	10.7	18.6	27.5	22.4	18.9%	7.6%	6.4%	7.4%	80.3%
ScotGrid	3.1	3.8	36.9	36.5	37.1	25.8%	24.4%	39.4%	41.2%	56.6%
SouthGrid	1.7	5.4	5.9	13.7	15.2	11.7%	3.5%	20.0%	4.8%	88.6%
NorthGrid	2.7	4.3	4.8	67.1	67.9	4.4%	6.7%	18.8%	2.1%	50.4%
Total Tier-2	**8.7**	**24.3**	**66.2**	**144.7**	**142.5**	**15.6%**	**9.2%**	**26.9%**	**13.2%**	**60.7%**
RAL Tier-1			**136.2**	**88.4**	**121.1**			**45.9%**	**50.0%**	**46.6%**

Table-4: LCG Disk Usage

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

OxGrid, a campus grid for the University of Oxford

David C. H. Wallom, Anne E Trefethen

Oxford e-Research Centre, University of Oxford,
7 Keble Road, Oxford OX2 6NN
david.wallom@oerc.ox.ac.uk

Abstract

The volume of computationally and data intensive research in a leading university can only increase. This though cannot be said of funding, so it is essential that every penny of useful work be extracted from existing systems. The University of Oxford has invested in creating a campus wide grid. This will connect all large scale computational resources as well as providing a uniform access method for 'external' resources such as the National Grid Service (NGS) and the Oxford Supercomputing Centre (OSC).

The backbone of the campus grid is made using standard middleware and tools but the value add services have been provided by in-house designed software including resource brokering, user management and accounting.

Since the system was first started in November 2005 we have attracted six significant users from five different departments. These use a mix of bespoke and licensed software and have run ~6300 jobs by the end of July 2006. Currently all users have access to ~1000 CPUs including NGS and OSC resources. With approximately 2 new users a week approaching the e-Research centre the current limitation on rate of uptake is the amount of time that is spent with each user to make their interactions as successful as possible.

1. Introduction

Within a leading university such as Oxford there could be expected to be as many as 30 separate clustered systems. These will have been purchased through a variety of sources such as grant income, donations, and central university funding. It is becoming increasingly important that full use is made of these. As well as these specialist systems, this is also true for all ICT resources throughout an organisation. These can include shared desktop computers within teaching laboratories as well as personal systems on staff desks.

There are also a significant number of resources that are available either nationally or internationally and it is therefore very important that the interfaces as defined by these projects are supported within the infrastructure. This has therefore a significant steer on the middleware chosen as the basis for the project. This it should be noted is true not only for computation but data as well.

The other large impediment that can occur for a project of this type is the social interactions between departments that may jealously guard their own resources and the users from different groups that could make best use of them. This is especially true in a collegiate university where you have possible resources that are located within separate colleges as well as the academic departments. This has therefore led to a large outreach effort through talks and seminars given to a university wide audience as well as contacting all serial users of the OSC.

The design of each of these components will be discussed showing their functional range as well as future plans to make further use of GGF standards.

2. Requirements

Before embarking on a exercise such as the construction of a campus wide infrastructure, it is important that an initial set of minimum user requirements are considered. The most important requirement is that the users current methods of working must be affected as little as possible, i.e. they should be able switch from working on their current systems to the campus grid with a seamless transition. The inherent system design should be such that its configuration can be dynamically altered without user interruption. This will include the interruption of service to particular clusters that make up nodes on the grid but also the central services. In this case the user may not be able to submit more tasks but should be safe in the knowledge that those tasks that they have already submitted will run uninterrupted. The final core requirement involves monitoring, since it is essential that once a user has submitted a job or stored a piece of data that it is monitored until its lifetime has expired.

2.1 Data provision as well as computation

The provision of data services will become increasingly important in the coming years. This will be especially true for the move by the arts, humanities and social sciences into e-Science, as these subjects include studies that make extensive use of data mining as a primary tool for research.

The data system must be able to take the following factors into account:

- Steep increase in the volume of data as studies progress including the new class of research that is generated by computational grid work.
- Metadata to describe the properties of the data stored, the volume of which will be directly proportional to its quality.
- As the class of data stored changes from final post analysis research data towards the raw data on which many different studies can be done then the need for replication and guaranteed quality of storage will increase.

Therefore a system is needed which can allow a range of physical storage medias to be added to a common core to present a uniform user interface. The actual storage must also be location independent different physical locations as possible.

3. The OxGrid System

The solution as finally designed is one where individual users interact with all connected resources through a central management system as shown in Figure 1. This has been configured such that as the usage of the system increases each component can be upgraded so as to allow organic growth of the available resources.

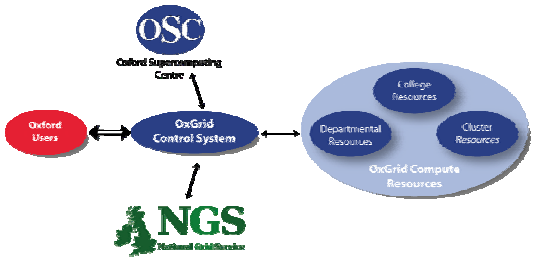


Figure 1 Schematic of the OxGrid system

3.1 Authorisation and Authentication

The projected users for the system can be split into two distinct groups, those that want to use only resources that exist within the university and those that want access to external systems such as the National Grid Service etc.

For those users requiring external system access, we are required to use standard UK e-Science Digital Certificates. This is a restriction bought about by those system owners.

For those users that only access university resources, a Kerberos Certificate Authority [3] system connected to the central university authentication system has been used. This has been taken up by central computing services and will therefore become a university wide service once there is a significant enough user base.

3.2 Connected Systems

Each resource that is connected to the system has to have a minimum software stack installed. The middleware installed is the Virtual Data Toolkit [4]. This includes the Globus 2.4 middleware [5] with various patches that have been applied through the developments of the European Data Grid project [6].

3.3 Central Services

The central services of the campus grid may be broken down as follows:

- Information Server
- Resource Broker
- Virtual Organisation Management
- Data Vault

These will all be described separately.

3.3.1 Information Server

The Information server forms the basis of the Campus Grid with all information about resources registered to it. Using the Globus MDS 2.x [7] system to provide details of the type of resource, including full system information. This also contains details of the installed scheduler and its associated queue system. Additional information is added using the GLUE schema [7].

3.3.2 Resource Broker

The Resource Broker (RB) is the core component of the system and the one with which the users of the system have the most interaction. Current designs for a resource broker are either very heavyweight with added functionality which is unlikely to be needed by the users or have a significant number of interdependencies which could make the system difficult to maintain.

The basic required functionality of a resource broker is actually very simple is listed below:

- Submit tasks to all connected resources,
- Automatically decide the most appropriate resource to distribute a task to,
- Depending on the requirements of the task submit to only those resources which fulfil them,
- Dependant on the users list of registered systems distribute tasks appropriately.

Using the Condor-G [9] system as a basis an additional layer was built to interface the Condor Matchmaking system with MDS. This involved interactively querying the Grid Index Information Service to retrieve the information stored in the system wide LDAP database as well as extracting the user registration information and installed software from the Virtual Organisation Management system.

Additionally other information such as which system a particular user is allowed to submit tasks

is available from the VOM system. Therefore this must be queried whenever a task is submitted.

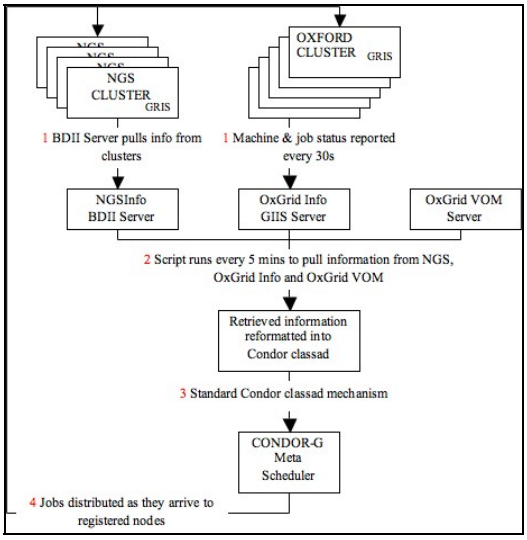


Figure 2 Resource Broker operation

3.3.2.1 The Generated Class advertisement

The information passed into the resource class advertisement may be classified in three ways, that which would be included in a standard Condor Machine class-ad, additional items that are needed for Condor-G grid operation and items which we have added to give extra functionality. This third set of information will be described here;

Requirements = (CurMatches < 20) && (TARGET.JobUniverse == 9)

It is important to ensure that a maximum number of submitted jobs can be matched to the resource at any one time and the resource will only accept Globus universe jobs.

CurMatches = 0

This is the number of currently matched jobs as determined by the advertisement generator using MDS and Condor queue information every 5 mins.

OpSys = "LINUX“
Arch = "INTEL"
Memory = 501

Information of the type of resource as is determined from the MDS information. It is important to note though that a current limitation is that this is for the head node only not workers so heterogeneous clusters cannot at present be used on the system.

MPI = False
INTEL_COMPILER=True
GCC3=True

Special capabilities of the resource need to be defined. In this case the cluster resource does not have MPI installed and so cannot accept parallel jobs. The list of installed software is also retrieved at this point from the Virtual Organisation Manager database from when the resources were added.
Each of these generated job advertisements for each resource are input into the Condor Matchmaking [10] system once every 5 mins.

3.3.2.2 Job Submission Script

As the standard user for the campus grid will have normally only submitted their jobs to a simple scheduler it is important that we can abstract users from underlying grid system and resource broker. The Condor-G system uses non-standard job description mechanisms and so a simpler more efficient method has been implemented. It was decided that most users were experienced with a command line executable that required arguments rather than designing a script submission type system. It is intended though to alter this with version 2 so that users may also use the GGF JSDL [11] standard to describe their tasks.
The functionality of the system must be as follows:

- User must be able to specify the name of the executable and whether this should be staged from the submit host or not,
- Any arguments that must this executable be passed to run on the execution host,
- Any input files that are needed to run and so must be copied onto the execution host,
- Any output files that are generated and so must be copied back to the submission host,
- Any special requirements on the execution host such as MPI, memory limits etc.
- Optional, so as to override the resource broker as necessary the user should also be able to specify the gatekeeper URL of the resource he specifically wants to run on. This is useful for testing etc.

It is important also that when a job is submitted that the user will not get his task allocated by the resource broker onto system to which he doesn't have access. The job submission script accesses the VOM system to get the list of allowed system for that user. This works through passing the DN into the VOM and retrieving a comma separated list of systems and reformats this into the format as accepted by the resource broker.

job-submission-script -n 1 -e /usr/local/bin/rung03 -a test.com -i test.com -o test.log -r GAUSSIAN03 -g maxwalltime=10

This example runs a Gaussian job, which in the current configuration of the grid will through the resource broker only run on the Rutherford NGS node. This was a specific case that was developed to test capability matching.

3.3.3 Virtual Organisation Manager

This is another example of a new solution being designed in-house due to over complicated solutions being only currently available. The functionality required is as follows:

- Add/Remove system to a list of available systems,
- List available systems,
- Add/Remove users to a list of users to access general systems,
- List users currently on the system.
- Add user to the SRB system,

When removing users though it is important that their record is set as invalid rather than simply removing entries so that system statistics are not disrupted.

So that attached resources can retrieve the list of stored distinguished names we have also created an LDAP database that can then be retrieved from via the standard EDG MakeGridmap scripts as distributed in VDT.

3.3.3.1 Storing the Data

The underlying functionality has been provided using a relational database. This has allowed the alteration of tables as and when additional functionality has become necessary. The decision was made to use the PostgreSQL relational Database [12] due to its know performance and zero cost option. It was also important at this stage to build in the extra functionality to support more than one virtual organisation. The design of the database is shown in Appendix 1.

3.3.3.2 Add, Remove and List Functions

Administration of the VOM system is through a secure web interface. This has each available management function as a separate web page which must be navigated to.

The underlying mechanisms for the addition, removal and list functions are the basically the same for both systems and users.

The information required for an individual user are:

- Name: The real name of the user.
- Distinguished Name (DN): This can either be a complete DN string as per a standard x509 digital certificate or if the Oxford Kerberbos CA is used then just their Oxford username, the DN for this type of user is constructed automatically.
- Type: Within the VOM a user may either be an administrator or user. This is used to define the level of control that he has over the VOM system, i.e. can alter the contents. This way the addition of new system administrators into the system is automated.
- Which registered systems the user can use and their local username on each, this can either be pool accounts or a real username.

An example of the interface is shown in Figure 3.

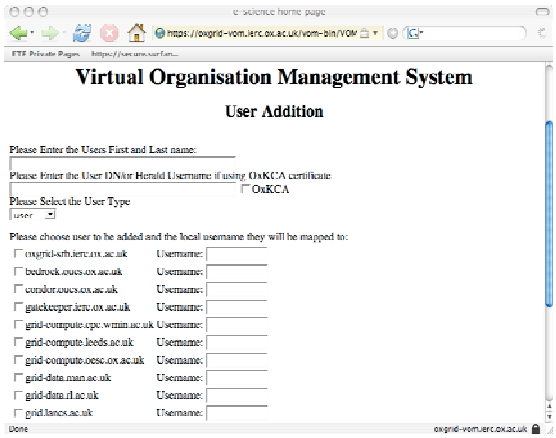


Figure 3; The interface to add a new user to the VOM

When a user is added into the VOM system this gives them access to the main computational resources through insertion of their DN into LDAP and relational databases. It is also necessary that each user is also added into the Storage Resource Broker [13] system for storage of their large output datasets.

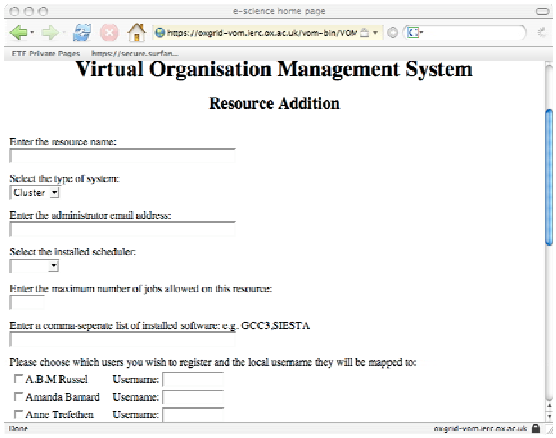


Figure 4; The interface for system registration into the VOM

To register a system with the VOM the following information is needed;

- Name: Fully Qualified Domain Name.
- Type: Either a Cluster or Central system, users will only be able to see Clusters.
- Administrator e-Mail: For support quereies.
- Installed Scheduler: Such as PBS, LSF, SGE or Condor.
- Maximum number of submitted jobs, to give the resource broker the value for 'CurMatches'.

- Installed Software; List of the installed licensed software to again pass onto the resource broker.
- Names of allowed users and their local usernames.

An example of the interface is shown in Figure 4.

3.4 Resource Usage Service

Within a distributed system where different components are owned by different organisations it is becoming increasingly important that tasks run on the system are accounted for properly.

3.4.1 Information Presented

As a base set of information the following should be recorded:

- Start time
- End time
- Resource name job run on
- Local job ID
- Grid user identification
- Local user name
- Executable run
- Arguments passed to the executable
- Wall time
- CPU time
- Memory used

As well as these basic variables an additional attribute has been set to account for the differing cost of a resource. This can be particularly used for systems that have special software or hardware installed. This can also be used to form the basis of a charging model for the system as a whole.

- Local Resource cost

3.4.2 Recording usage

There are two parts to the system, a client that can be installed onto each of the attached resources and a server that will record the information for presentation to system administrators and users. It was decided that it would be best to present the accounting information to the server on a task by task basis. This would result in instantaneously correct statistics should it be necessary to apply limits and quotas. The easiest way to achieve this is through the creation of a set of Perl library functions that attaché to the standard job-managers that are distributed within VDT. These collate all of the information to be presented to the server and then call a 'cgi' script through a web interface on the server.

The server database is part of the VOM system described in section 3.3.3.1. An extra table has been added with each attribute corresponding to a column and each recorded task is a new row.

3.4.3 Displaying Usage

There are three different methods of displaying the information that is stored about tasks run on OxGrid. The overall number of tasks per month is shown in Figure 5.

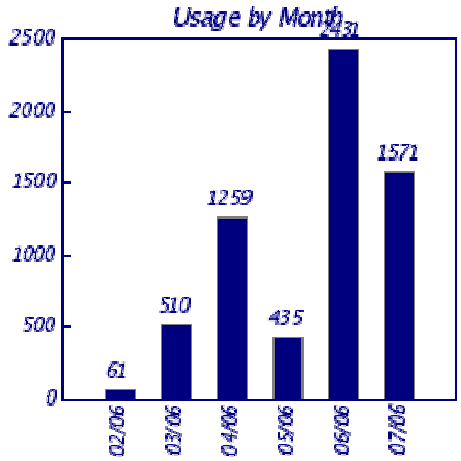


Figure 5; Total number of run tasks per month on all connected resources owned by Oxford, i.e. NOT including remote NGS core nodes or partners.

This can then be split into total numbers per individual user and per individual connected system as shown in Figure 6 and Figure 7.

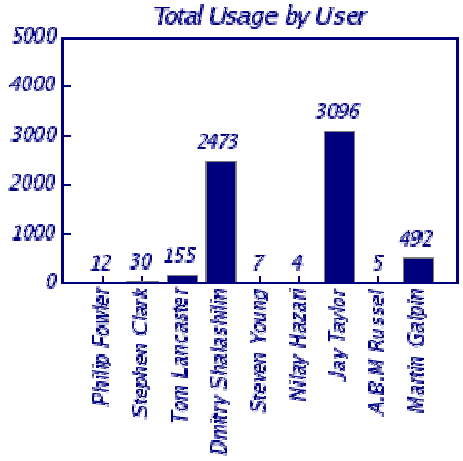


Figure 6; Total number of submitted tasks per user

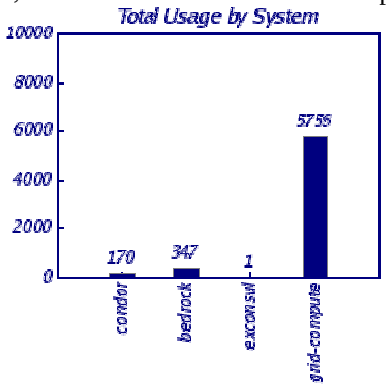


Figure 7; Total number of jobs as run on each connected system

3.5 Data Storage Vault

It was decided that the best method to create an interoperable data vault system would be to leverage work already undertaken within the UK e-Science community and in particular by the NGS. The requirement is for a location independent virtual filesystem. This can make use of spare diskspace that is inherent in modern systems. Initially though as a carrot to attract users we have added a 1Tb RAID system onto which users data can be stored. The storage system uses the Storage Resource Broker (SRB) from SDSC. This has the added advantage of not only fulfilling the location independent requirement but can also add metadata to annotate stored data for improved data mining capability.

The SRB system is also able to interface not only to plain text files that are stored within normal attached filesystems but also relational databases. This will allow large data users within the university to make use of the system as well as install the interfaces necessary to attach their own databases with minimal additional work.

3.6 Attached Resources

When constructing the OxGrid system the greatest problem that was encountered was with the different resources that have been connected to the system. These fell into separate classes as described.

3.6.1 Teaching Lab Condor Pools

The largest single donation of systems into the OxGrid has come from the Computing Services teaching lab systems. These are used during the day as Windows systems and have a dual boot installation setup on them with a minimal Linux 2.4.X installation on them. The systems are rebooted into Linux at 2100 each night and then run as part of the pool until 0700 where they are restarted back into Windows systems. Problems have been encountered with the system imaging and control software used by OUCS and its Linux support. This has lead to significant reduction in available capacity in this system until the problem is rectified. Since this system also is configured without a shared filesystem several changes have been made to the standard Globus jobmanager scripts to ensure that Condor file transfer is used within the pool.
A second problem has been found recently with the discovery of a large number of systems within a department that use the Ubuntu Linux distribution which is currently not supported by Condor. This has resulted in having to distribute by hand a set of base C libraries that solve issues with running the Condor installation

3.6.2 Clustered Systems

Since there is significant experience within the OeRC with clustered systems we have been asked on several occasions to assist with cluster upgrades before these resources can be added into the campus grid. This has illustrated the significant problems with the Beowulf solution to clustering, especially if very cheap hardware has been purchased. This has lead on several occasions to the need to spend a significant amount of time installing operating systems which is reality has little to do with construction of a campus grid.

3.6.3 Sociological issues

The biggest issue when approaching resource owners is always the initial reluctance to allow anyone but themselves to use resources they own. This is a general problem with academics all over the world and as such can only be rectified with careful consideration of their concerns and specific answers to the questions they have. This has resulted in a set of documentation that can be given to owners of clusters and Condor systems so that they can make informed choices on whether they want to donate resource or not.

The other issue that has had to be handled is the communication with staff responsible for departmental security. To counteract this we have produced a standard set of requirements with firewalls which are generally well received. By ensuring that communication from departmental equipment is a single system for the submission of

3.7 User Tools

In this section various tools are described that have been implemented to make user interaction with the campus grid easier.

3.7.1 oxgrid_certificate_import

One of the key problems within the UK e-Science community has always been that users have complained about difficult interactions with their digital certificates and certainly when dealing with the format required by the GSI infrastructure. It was therefore decided to produce an automated script. The is used to translate the certificate as it is retrieved from the UK e-Science CA, automatically save it in the correct location for GSI as well as set permissions and the passphrase used to verify the creation of proxy identities. This has been found to reduce the number of support calls about certificates as these can cause problems which the average new user would be unable to diagnose. To ensure that the operation has completed successfully it also checks the creation of a proxy and prints its contents.

3.7.2 oxgrid_q and oxgrid_status

So that a user can check their individually submitted tasks we developed a script that sits on top of the standard condor commands for showing the job queue and registered systems. Both of these commands though as default show only the jobs

the user has submitted or the systems they are allowed to run jobs on, though both of these commands also have global arguments to allow all jobs and registered systems to be viewed.

3.7.3 oxgrid_cleanup

When a user has submitted many jobs and discovered that they have made an error in configuration etc. it is important for a large distributed system such as this that a tool exists for the easy removal of not just the underlying submitted tasks but also the wrapper submitter script as well. Therefore this tool has been created to remove the mother process as well as all its submitted children.

4. Users and Statistics

Currently there are 25 registered users of the system. They range from a set of students who are using the submission capabilities to the National Grid Service to those users that are only wanting to use the local Oxford resources.
They have collectively run ~6300 tasks over the last six months using all of the available systems within the university (including the OSC), not including though the rest of the NGS.
The user base is currently determined by those that have been registered with the OSC or similar organizations before. Through outreach efforts though this is being moved into the more data intensive Social Sciences as their e-Science projects move along. Several whole projects have also registered their developers to make use of the large data vault capability including the Integrative Biology Virtual Research Environment.
We have had several instances where we have asked for user input and a sample of them are presented here.
"**My work is the simulation of the quantum dynamics of correlated electrons in a laser field. OxGrid made serious computational power easily available and was crucial for making the simulating algorithm work."** *Dr Dmitrii Shalashilin (Theoretical Chemistry)*
"The work I have done on OxGrid is on molecular evolution of a large antigen gene family in African trypanosomes. OeRC/OxGrid has been key to my research and has allowed me to complete within a few weeks calculations which would have taken months to run on my desktop." *Dr Jay Taylor* (Statistics)

5. Costs

There are two ways that the campus grid can be valued. Either in comparison with a similar sized cluster or in the increased throughput of the supercomputer system.
Considering the 250 systems of the teaching cluster can be the basis for costs. The electricity costs for these systems would be £7000 if they were left turned on for 24 hours. These systems though do produce 1.25M CPU hours of processing power so the produced value of the resource is very high.
Since the introduction of the campus grid it is considered that the utilisation of the supercomputing centre has increased for MPI tasks by ~10%.

6. Conclusion

Since November 2005 the campus grid has connected systems from four different departments including Physics, Chemistry, Biochemistry and University Computing Services. The resources located in the National Grid Service and OSC have also been connected for registered users. User interaction with these physically dislocated resources is seamless when using the resource broker and for those under the control of the OeRC accounting information has been saved for each task run. This has showed that ~6300 jobs have been run on the Oxford based components of the system (i.e. not including the wider NGS core nodes or partners).
The added advantage is that significant numbers of serial users from the Oxford Supercomputing Centre have moved to the campus grid so that it has also increased its performance.

7. Acknowledgments

I would like to thank Prof. Paul Jeffreys and Dr. Anne Trefethen for their continued support in the startup and construction of the campus grid. I would also like to thanks Dr. Jon Wakelin for his assistance in the implementation and design of some aspects of the Version 1 of the underlying software when the author and he were both staff members at Centre for e-Research Bristol.

8. References

1. National Grid Service, http://www.ngs.ac.uk
2. Oxford Supercomputer Centre, http://www.osc.ox.ac.uk
3. Kerberos CA and kx509, http://www.citi.umich.edu/projects/kerb_pki/
4. Virtual Data Toolkit, http://www.cs.wisc.edu/vdt
5. Globus Toolkit, The Globus Alliance, http://www.globus.org.
6. European Data Grid, http://eu-datagrid.web.cern.ch/eu-datagrid/
7. Cza jkowski k., Fitzgerald s., Foster I., and Kesselman C. Grid Information Services for Distributed Resource Sharing Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.

8. GLUE schema The Grid Laboratory Uniform Environement (GLUE)", http://www.hicb.org/glue/glue.htm. Computing (HPDC-10), IEEE Press, August 2001.

9. J. Frey, T. Tannenbaum, M. Livny, I. Foster, S. Tuecke, Condor-G: A Computation Management Agent for Multi-Institutional Grids, Cluster Computing, Volume 5, Issue 3, Jul 2002, Pages 237 – 246

10. R. Raman, M. Livny, M. Solomon, "Matchmaking: Distributed Resource Management for High Throughput Computing," hpdc, p. 140, Seventh IEEE International Symposium on High Performance Distributed Computing (HPDC-7 '98), 1998.

11. Job Submission Description Language (JSDL) Specification, Version 1.0, GGF, https://forge.gridforum.org/pro jects/jsdl-wg/document/draft-ggf-jsdl-spec/en/21

12. PostgreSQL, http://www.postgresql.org

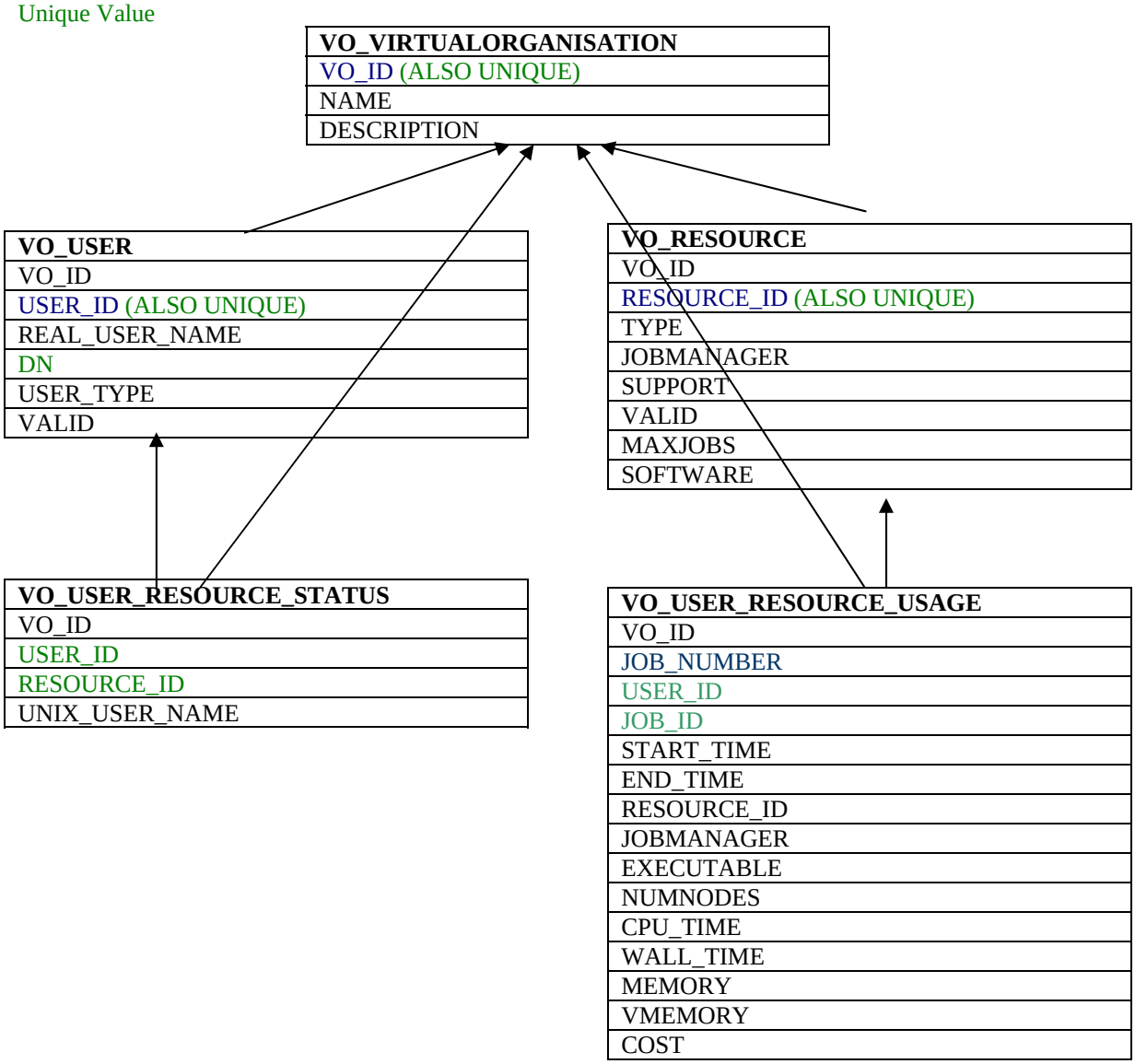
13. C. Baru, R. Moore, A. Rajasekar and M. Wan. The SDSC storage resource broker. In Proceedings of the 1998 Conference of the Centre For Advanced Studies on Collaborative Research (Toronto, Ontario, Canada, November 30 - December 03, 1998)

14. LHC Computing Grid Project http://lcg.web.cern.ch/LCG/

Appendix 1

Table structure of the VOM database system.

Table Primary Key
Unique Value



Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

GridPP: Running a Production Grid

Olivier van der Aa[1], **Stephen Burke**[2], Jeremy Coles[2], David Colling[1], Yves Coppens[3], Greig Cowan[4], Alessandra Forti[5], Peter Gronbech[6], Jens Jensen[2], David Kant[2], Gidon Moont[1], Fraser Speirs[7], Graeme Stewart[7], Philippa Strange[2], Owen Synge[2], Matt Thorpe[2], Steve Traylen[2]

[1]Imperial College London [2]CCLRC [3]University of Birmingham [4]University of Edinburgh [5]University of Manchester [6]University of Oxford [7]University of Glasgow

Abstract

GridPP operates a Grid for the UK particle physics community, which is tightly integrated with the Enabling Grids for E-sciencE (EGEE) and LHC Computing Grid (LCG) projects. These are now reaching maturity: EGEE recently entered its second phase, and LCG is ramping up to the start of full-scale production in 2007, so the Grid must now be operated as a full production service. GridPP provides CPU and storage resources at 20 sites across the UK, runs the UK-Ireland Regional Operations Centre for EGEE, provides Grid-wide configuration, monitoring and accounting information via the Grid Operations Centre, and takes part in shift rotas for Grid operations and user support. It also provides support directly for its own system managers and users. This paper describes the structure and operations model for EGEE and LCG and the role of GridPP within that, focusing on the problems encountered in operating a worldwide production Grid with 24/7 availability, and the solutions which are being developed to enable this to succeed.

1. EGEE and LCG

Enabling Grids for E-sciencE (EGEE) [1] is an EU project to develop a large e-science Grid in Europe, which also has extensive links with projects in other countries. The first phase started in April 2004 and completed successfully in March this year. The second phase runs from April 2006 to March 2008, and discussions are currently underway concerning the future of the infrastructure beyond that date. EGEE is intended to serve the entire scientific community, although particle physics is currently the largest user.

The LHC Computing Grid (LCG) project [2] will provide the computing infrastructure for the Large Hadron Collider (LHC), which is due to start operation in 2007. Much of its infrastructure is provided within the EGEE framework, but it also needs to interoperate with other Grids, notably the US Open Science Grid (OSG) [3].

The combined EGEE/LCG Grid now consists of over 200 sites in around 40 countries (see Figure 1), provided by a large number of affiliated national and regional Grid projects, with in excess of 25,000 CPUs and 12 Pb of storage. There are well over 2000 registered users, organized into nearly 200 Virtual Organisations (VOs). Managing such a large, distributed infrastructure is a non-trivial problem, and the projects have been working to develop tools and procedures to provide a production-quality service.

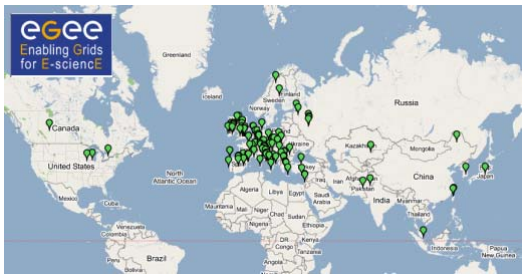


Figure 1: Sites which are part of the EGEE/LCG Grid, displayed on a Google map

1.1 EGEE structure

The EGEE project is split into a number of Activities in three broad areas: operations (SA), middleware development (JRA), and interactions with users and external bodies (NA). The SA1 Activity is responsible for Grid deployment and operations, and GridPP [4] is a major contributor to it. Overall co-ordination of deployment and operations is provided by the Operations Co-ordination Centre (OCC) at the CERN laboratory in Geneva.

EGEE is organised into regions, each of which has a Regional Operations Centre (ROC). The function of the ROC is to provide deployment and operational support for the sites in their region, to contribute to user support, and to represent the region to EGEE as a whole. CERN acts as a ROC for those sites without a regional affiliation.

GridPP is part of the UK-Ireland (UK-I) region, and runs the UK-I ROC. UK-I resources

currently come largely from GridPP and GridIreland [5], but we also anticipate some degree of convergence with the National Grid Service (NGS) [6], and we are currently seeing some interest from other e-science sites in the UK.

GridPP also co-ordinates the Grid Operations Centre (GOC) [7]. This maintains a database of information about all the sites in the Grid and provides a variety of monitoring and accounting tools.

In EGEE-1 a subset of the ROCs were classed as Core Infrastructure Centres (CICs). These were involved in running critical services and taking on a rotating responsibility to monitor the state of the Grid and take action to solve problems. In EGEE-II the CIC functionality is being absorbed into the ROCs as more regions gain the necessary expertise. GridPP has been involved with the CIC functions from the start.

1.2 LCG structure

LCG is dedicated to providing for the computing needs of the experiments at the LHC particle accelerator. These are expected to produce several Pb of primary data per year, together with similar volumes of derived and simulated data, so data movement and storage is therefore a major consideration.

LCG organises its sites into a hierarchical, tiered structure. Tier 0 is at CERN, the laboratory which hosts the LHC accelerator. Files are shipped from there to a number of Tier 1 sites, which have substantial CPU, storage and manpower resources, high-capacity network connections, can guarantee high availability, and are able to make a long-term commitment to supporting the experiments. Tier 1 sites will also need to transfer data between themselves.

Each Tier 1 site supports a number of Tier 2 sites, which are allowed to have lower levels of resources and availability. Files are expected to be transferred in both directions between Tier 2 sites and their associated Tier 1, but generally not on a wider scale, although the detailed data movement patterns are still under discussion.

LCG is putting in place Service Level Agreements (SLAs) to define the expected performance of the Tier centres, and is also developing monitoring tools to measure actual performance against the SLAs. A set of Service Challenges are being run, principally to test the routine transfer of large data volumes but also to test the full computing models of the LHC experiments.

In the UK, GridPP has a Tier 1 site at the Rutherford Appleton Laboratory [8] and 19 Tier 2 sites at Universities around the country [9]. For administrative convenience these are grouped into four “virtual Tier 2s”. From a technical point of view each site operates independently, but the Tier 2s each have an overall co-ordinator and are moving towards some sharing of operational cover.

1.3 Virtual Organisations

Users are grouped into Virtual Organisations (VOs), and sites can choose which VOs to support. EGEE is developing formal policies under which VOs can be approved, but at present there are effectively several different categories of VO:

- LCG-approved VOs support the four LHC experiments, and also some associated activities.
- EGEE-approved VOs have gone through a formal induction process in the EGEE NA4 (application support) Activity, and receive support from EGEE to help them migrate to the Grid. However, this is a relatively heavyweight process and so far rather few VOs have come in via this route.
- Global VOs have a worldwide scope, but have not been formally approved by EGEE and do not receive direct support from it. At present these are predominantly related to particle physics experiments.
- Regional VOs are sponsored, and largely supported, by one of the participating Grid projects. The UK-I ROC currently has a small number of VOs in this category, but the number is expected to increase. Regional VOs may also be supported at sites outside the region if they have some relationship with the VO.

Although the status of the VOs varies, as far as possible they are treated equivalently from a technical point of view in both the middleware and the infrastructure. Each VO has a standard set of configuration parameters, which makes it relatively easy for a site to add support for a new VO (unless there are special requirements).

As a particle physics project GridPP is mainly interested in supporting VOs related to the experiments in which the UK participates. However, it has a policy of providing resources at a low priority for all EGEE-approved VOs.

2. Middleware

The EGEE/LCG middleware is in a state of constant evolution as the technology develops, and managing this evolution while maintaining

a production service is a major component of Grid deployment activity. The current middleware is based on software from Globus [10] (but still using the Globus toolkit 2 release dating from 2002), Condor [11] and other projects, as collected into the Virtual Data Toolkit (VDT) [12]. A substantial amount of the higher-level middleware came from the European DataGrid (EDG) project [13], the predecessor of EGEE. Many other tools have been developed within LCG. The middleware Activity in EGEE (JRA1) has developed new software under the gLite brand name, and an integrated software distribution has recently been deployed on the production system which has adopted the gLite name [14].

The middleware can broadly be split into three categories: site services which are deployed at every site, core services which provide more centralised functions, and Grid operation services which relate to the operation of the Grid as a whole.

2.1 Site services

Every site needs to deploy services related to processing, data storage and information transport. There has also recently been some discussion about providing facilities for users to deploy their own site-level services. The individual services are as follows:

- Berkeley Database Information Index (BDII) – this is an LDAP server which allows sites to publish information according to the GLUE information schema [15].
- Relational Grid Monitoring Architecture (R-GMA) – another information system presenting information using a relational model. This is also used to publish the GLUE schema information, together with a variety of monitoring and accounting information, and is also available for users to publish their own data.
- Computing Element (CE) – this provides an interface to computing resources, typically via submission to a local batch system. This has so far used the Globus gatekeeper, but the system is now in transition to a new CE interface based on Condor-C. A tool called APEL (Accounting Processor using Event Logs) is also run on the CE to collect accounting information, and transmit it via R-GMA to a repository at the GOC.
- Worker Nodes (WN) – these do not run any Grid services (beyond standard batch schedulers), but need to have access to the client tools and libraries for Grid services which may be needed by running jobs, and often to VO-specific client software.
- Storage Element (SE) – this provides an interface to bulk data storage. Historically the so-called "classic SE" was essentially just a Globus GridFTP server, but the system is now moving to a standard interface known as the Storage Resource Manager (SRM) [16].
- VOBOX – there have recently been requests to have a mechanism to deploy persistent services on behalf of some VOs. This is currently under discussion due to concerns about scalability and security, but some such services are currently deployed on an experimental basis.

2.2 Core services

Some services are not needed at every site, but provide some general Grid-wide functionality. Ideally it should be possible to have multiple instances of these to avoid single points of failure, although this has not yet always been achieved in practice. The services are:

- Resource Broker (RB) – this accepts job submissions from users, matches them to suitable sites for execution, and manages the jobs throughout their lifetime. A separate Logging and Bookkeeping (LB) service is usually deployed alongside the broker.
- MyProxy [17] – a repository for long term user credentials, which can be contacted by other Grid services to renew short-lived proxies before they expire.
- BDII – the information from the site-level BDII servers is collected into Grid-wide BDIIs which provide a view of the entire Grid. BDIIs are usually associated with RBs to provide information about the available resources to which jobs can be submitted.
- R-GMA Registry and Schema – these are the central components of the R-GMA system, providing information about producers and consumers of data and defining the schema for the relational tables. At present there is a single instance for the whole Grid.
- File Catalogue – this allows the location of file replicas stored on SEs using a Logical FileName (LFN). The system is currently in transition from the EDG-derived Replica Location Service (RLS) to the LCG File Catalogue (LFC) which has improved

performance. At present this is generally deployed as a single instance per VO, but the system is designed to allow distributed deployment.

- File Transfer Service (FTS) – this provides a reliable, managed service to transfer files between two SEs.
- User Interface (UI) – this has the various client tools to allow jobs to be submitted and monitored, files to be managed, R-GMA to be queried etc. This can be a shared service or installed on a user's own machine.
- VO server – this stores the membership information for a VO. The system is currently in transition from a solution based on LDAP servers to a more powerful system called the VO Membership Service (VOMS). At present these servers are normally deployed as a single instance per VO, although there is some scope for redundancy.

2.3 Grid operation services

It has proved to be necessary to develop some infrastructure services/tools to support the operation of the Grid as a whole. These generally relate to testing and monitoring the system:

- The GOC Database (GOCDB) has information about every site in the system, including its status, contact details, and a full list of nodes. This is used by various other tools, e.g. to generate lists of sites/nodes to monitor. The information is also republished into R-GMA.
- The operations portal [18] is a web site which collects a variety of operational information and tools for users, VO managers, site and ROC managers, and operations staff.
- The Global Grid User Support (GGUS) portal [19] is a web site which enables support tickets to be entered and tracked, and also has links to a wide variety of user support information and documentation.
- The Site Functional Tests (SFTs) [20] are test jobs which run regularly at each site and test the major Grid functionality. The results are reported back to a central database, and can be displayed in various formats (see Figure 2). They can be used both to flag problems at the sites, and to measure the availability and reliability of sites and services. This is currently being upgraded to a new system known as SAM (Site Availability Monitoring).
- The Grid status monitor (GStat) [21] reads information published by each site according to the GLUE schema, and uses this to display the current status and history, and flag potential problems.
- The GridView tool collects file transfer information, published from GridFTP into R-GMA, and displays it.
- Freedom of Choice for Resources (FCR) is a tool which allows the selection of sites seen by an RB through a BDII to be customised dynamically. In particular, sites which are failing either standard or VO-specific tests can be removed, and individual sites can also be white- or black-listed.
- The Real Time Monitor (RTM) [22] collects information about jobs from LBs. This can be used to display the state of the system in real time, and also allows historical data to be processed to derive statistics on such things as job volumes, duration and failure rates.
- GridICE [23] displays monitoring information published both in the GLUE schema and in GridICE-specific extensions.

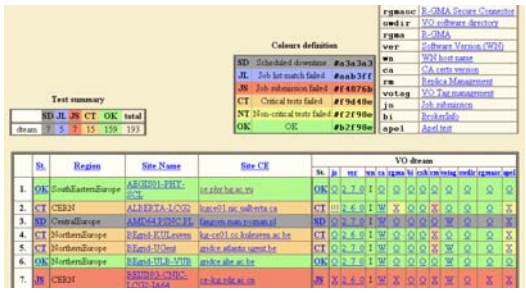


Figure 2: Part of the Site Functional Test display

2.4 Deployment issues

Middleware components are often developed by relatively small groups, with a single development platform and with installation and testing on a small system which is under their complete control. They are also typically developed as standalone services, rather than being designed from the start to work with other components in an integrated way. This can lead to the emergence of a number of problems when the middleware is deployed on a large-scale, heterogeneous system.

A basic issue is scalability; many components encounter problems coping with the large number of sites/jobs/VOs/users/files in the production system, and with continuous

operation over long periods. Unfortunately this is effectively confronted only on the production system, as test systems are too small to detect the problems.

It is often very difficult to port middleware to new operating systems (or even new versions of a supported operating system), compilers and batch systems. EGEE has an objective of supporting a wide range of platforms, but so far this has not been achieved in practice, and porting remains a painful exercise.

The middleware typically depends on a number of open-source tools and libraries, and the consequent package dependencies can be difficult to manage. This is particularly true when the dependency is on a specific version – which can lead to situations where different middleware components depend on different tool versions. This is partly a result of poor backward-compatibility in the underlying tools. These problems can be solved, but absorb a significant amount of time and effort.

In a large production Grid it is not practical to expect all sites to upgrade rapidly to a new release, so the middleware needs to be backward-compatible at least with the previous release and ideally the one before that. Any new services need to be introduced in parallel with the existing ones. This has generally been achieved, but is sometimes compromised by relatively trivial non-compatible changes.

The most important issue is configuration. Much of the middleware is highly flexible and supports a large number of configuration options. However, for production use most of this flexibility needs to be hidden from the system managers who install it, and Grid deployment therefore involves design decisions about how the middleware needs to be configured. There is also a scalability aspect, for example services should be discovered dynamically wherever possible rather than relying on static configuration. The middleware should also be robust in the sense that small changes to the configuration should not prevent it working, which is not always the case. In practice, finding a suitable configuration and producing a well-integrated release is a major task, which can take several months.

2.5 Operational issues

A production Grid requires continuous, reliable operation on a 24/7 basis. In a large system there are always faults somewhere, which may include hardware failures, overloaded machines, misconfigurations, network problems and host certificate expiry. Middleware is often written under the assumption that such errors are exceptional and must be rectified for the software to function. In a large Grid such "exceptions" are in fact the norm, and the software needs to be able to deal with this. In addition, error reporting is often regarded as having a low priority, and consequently it can be very hard to understand the reason for a failure. Ideally:

- Middleware should be fault tolerant; wherever possible it should retry or otherwise attempt a recovery if some operation fails.
- Services should be redundant, with automatic failover.
- Logging and error messages should be sufficient to trace the cause of any failure.
- It should be possible to diagnose problems remotely.

In practice these conditions are usually not fulfilled in the current generation of middleware, and it is therefore necessary to devise operational procedures to mitigate the effects of failures.

3. Grid deployment

Deployment relates to the packaging, installation and configuration of the Grid middleware.

3.1 Middleware installation and configuration

Various installation/configuration tools have been developed in a number of Grid projects which allow much of the middleware flexibility to be exposed in the configuration. However, as mentioned above, in practice system managers prefer to have the vast majority of configuration flexibility frozen out. LCG has therefore developed a tool called YAIM (Yet Another Installation Method) which uses shell scripts driven from a simple configuration file with a minimal number of parameters.

3.2 Middleware releases

Release management is a difficult area in a situation where the middleware continues to evolve rapidly. So far there have been major production releases about three times a year, with all the software being upgraded together. (This excludes minor patches, which are released on an ad-hoc basis.) However, this has caused some difficulties. On one side it takes a long time for new functionality, which may be urgently needed by some users, to get into the

system. On the other hand, this is still seen as too rapid by many sites. Practical experience is that it takes many weeks for most sites to upgrade to a new release, with some sites taking several months.

Deployment is now moving towards incremental upgrades of individual services to try to alleviate this situation. However, this may bring problems of its own, as the backward-compatibility requirements may become much greater with a more complex mix of service versions at different sites. The introduction of a Pre-Production System (PPS) to try out pre-release software should however enable more problems to be found before new software goes into full production.

Support for release installation is provided principally on a regional basis; GridPP has a deployment team which provides advice and support to sites in the UK.

3.3 Documentation and training

Documentation is needed for both system managers and users. Unfortunately there is little or no dedicated effort to produce it, and the rapid rate of change means that documentation which does exist is often out of date. Documentation is also located on a wide range of web sites with little overall organisation. However, more effort is now being made to improve this area.

For system managers the central deployment team provide installation instructions and release notes. There is also a fairly extensive wiki [24] which allows system managers to share their experiences and knowledge.

The biggest problem with user documentation is a lack of co-ordination. The main User Guide is well-maintained, but in other areas it relies on the efforts of individual middleware developers. Many user-oriented web sites exist, but again with little co-ordination. GridPP has its own web site with a fairly extensive user area [25], which attempts to improve the situation rather than add to the confusion by providing structured pointers to documentation held elsewhere. EGEE has recently set up a User Information Group (UIG) with the aim of co-ordinating user documentation, based around a set of specific use-cases.

Training within EGEE is provided by the NA3 Activity, which is co-ordinated by NESC [26]. The training material is also available on the web. However, again there is a problem with keeping up with the rate of change of the system, and also to some extent with reaching the highly diverse community of people who might require training.

One especially difficult area is the question of the induction of new VOs, to allow them to understand how to adapt their software and methods to make best use of the Grid. EGEE is intending to run periodic User Forum events at which users can share their experiences to improve the transfer of such information, with the first having been held in March this year [27]. In GridPP, so far most users have been from the large LCG VOs who have a fairly well-developed engagement with the Grid, but we will need to develop induction and training for internal VOs which may only have a small number of members.

3.4 VO support at sites

EGEE is intended to support a large number of VOs, and at present there are getting on for 200 VOs enabled somewhere on the infrastructure. This implies that it should be easy for a site to configure support for a new VO. However, there are currently some problems with this:

- The key configuration information is not always available. The operations portal has templates to allow VOs to describe themselves, but in most cases this is not yet in use.
- Supporting a new VO implies changes in various parts of the configuration, so this needs to be done in a parameterised way to make it as simple as possible - in the ideal case a site should simply have to add the VO name to a configuration file. YAIM now incorporates a web-based tool to generate the required configuration for those sites (the majority) which use it, which goes a long way towards this goal. Unfortunately, at present this also suffers from the fact that the configuration information for many VOs is not yet present.
- Some VOs may require non-standard configurations or extra services, and this decreases the chances that they will be supported by sites which do not have a strong tie to the VO. This is partly a question of user education; VOs need to realise that working within a Grid environment can accommodate less site-specific customisation than when working with a small number of sites. It may also imply the development of a standardised way to deploy non-standard services, the "VO box" concept mentioned above.

- Sites, and the bodies which fund them, may also need to adapt their attitude somewhat to the Grid model. Sites are often funded, and see themselves, as providing services for a small number of specific VOs, rather than a general service to the Grid community as a whole.

3.5 VO software installation

VOs often need to pre-install custom client tools and libraries which will be available to jobs running on WNs. The current method, which has been found to work reasonably well, is to provide an NFS-shared area identified by a standard environment variable, which is writeable by a restricted class of VO members. These people can also use GridFTP to write version tags to a special file on the CE, which are read and published into the information system, allowing jobs to be targeted at sites which advertise particular software versions. This tagging method can also be used to validate any other special features of the execution environment which are required by the VO.

4. Grid operations

As discussed above, failures at individual sites are a fact of life in such a large system, and the middleware currently does not deal with errors very effectively in most cases. The "raw" failure rate for jobs can therefore be rather high. In the early days it was also not unusual for sites to appear as "black holes": some failure modes can result in many jobs being sent to a site at which they either fail immediately, or are queued for a long period.

To get to a reasonable level of efficiency (generally considered to be a job failure rate below 5-10%) it has been necessary to develop operational procedures to mask the effects of the underlying failures. The key tool for this is the SFT system, which probes the critical functionality at each site on a regular basis. Failing sites can be removed from being matched for job submission, using criteria defined separately for each VO, using the FCR tool. In parallel with this an operations team submits tickets to the sites to ensure that problems are fixed as quickly as possible, as described below. All these systems and tools continue to be developed as we gain more experience. Performance is measured by a variety of tools, e.g. the RTM, which allow the current situation and trends over time to be monitored.

As mentioned earlier, LCG uses resources from other Grids, notably OSG in the US, and efforts are currently underway to integrate these into the EGEE/LCG operations framework as far as possible.

4.1 Site status

New sites are initially entered into the GOC DB as uncertified. The ROC to which they belong is expected to perform basic tests before marking the site as certified, at which point it becomes a candidate for regular job submission, and starts to be covered by the routine monitoring described below. Sites can also be marked as being in scheduled downtime, which temporarily removes them from the system.

4.2 Regular monitoring

An operations rota has been established to deal with site-level problems, with ROCs (formerly just CICs) being on duty for a week at a time. While on duty the various monitoring tools are scanned regularly (information is aggregated for this purpose on the operations portal). If problems are found tickets are opened against the relevant sites, and followed up with a specified escalation procedure, which in extreme cases can lead to sites being decertified. At present, with around 200 sites there are about 50 such tickets per week, which illustrates both that active monitoring is vital to maintaining the Grid in an operational state, and that there is never a time when the system is free of faults.

Tickets are initially opened in the GGUS system, but each ROC has its own internal system which it uses to track problems at the sites it serves, using a variety of technologies. An XML-based ticket exchange protocol has therefore been developed to allow tickets to be exchanged seamlessly between GGUS and the various regional systems.

Summaries of the operations activity are kept on the CIC portal, and issues are discussed at a weekly operations meeting.

4.3 Accounting

Accounting information is collected by the APEL tool and published to a database held at the GOC [28]. At present this only relates to the use of CPU time, but storage accounting is in development. Accounting information is currently only available for entire VOs (see Figure 3) due to concerns about privacy, but user-level information is likely to be required in the future once the legal issues are clarified. Accounting information is also available per site and with various degrees of aggregation above

that. So far accounting data is only used for informational purposes, but it is likely that it will be used in a more formal way in future, e.g. to monitor agreed levels of resource provision.

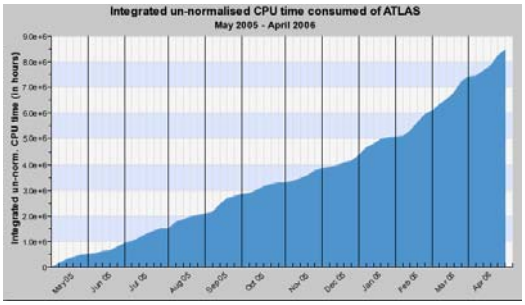


Figure 3: Accounting - integrated CPU time used by the atlas VO

4.4 User support

The question of user support initially received relatively little attention or manpower, but it has become clear that this is a very important area, and in the last year or so a substantial amount of effort has been put into developing a workable system. This is again based on the GGUS portal, where tickets can be entered directly or via email. People known as Ticket Processing Managers (TPMs) then classify the tickets, attempt to solve them in simple cases, or else assign them to the appropriate expert unit, and also monitor them to ensure that they progress in a timely way. TPMs are provided on a shift rota by the ROCs, and GridPP contributes to this, as well as providing some of the expert support units.

At present the system is working reasonably well, with typically a few tens of tickets per week. However, as usage of the system grows the number of tickets is also likely to grow, and it is not clear that the existing system can scale to meet this. The biggest problem is not technical, but the lack of manpower to provide user support, especially given that supporters need to be experts who generally have many other tasks. In the longer term we will probably need to employ dedicated support staff.

5. Summary

GridPP runs the UK component of the worldwide EGEE/LCG Grid. This is a very large system which is now expected to function as a stable production service. However, the underlying Grid middleware is still changing rapidly, and is not yet sufficiently robust. It has therefore been vital to develop Grid operations procedures which allow users to see only those parts of the system which are working well at any given time, and to manage software upgrades in a way which does not disrupt the service. This is still being developed but is now working well, and usage of the Grid has been rising steadily. Many challenges nevertheless remain, especially as the LHC starts to take data.

References

[1] http://www.eu-egee.org/
[2] http://lcg.web.cern.ch/LCG/
[3] http://www.opensciencegrid.org/
[4] http://www.gridpp.ac.uk/
[5] http://cagraidsvr06.cs.tcd.ie/index.html
[6] http://www.ngs.ac.uk/
[7] http://goc.grid-support.ac.uk/gridsite/gocmain/
[8] http://www.gridpp.ac.uk/tier1a/
[9] http://www.gridpp.ac.uk/tier2/
[10] http://www.globus.org/
[11] http://www.cs.wisc.edu/condor/
[12] http://vdt.cs.wisc.edu/
[13] http://www.edg.org/
[14] http://www.glite.org/
[15] http://infnforge.cnaf.infn.it/glueinfomodel/
[16] http://sdm.lbl.gov/srm-wg/
[17] http://grid.ncsa.uiuc.edu/myproxy/
[18] http://cic.in2p3.fr/
[19] http://www.ggus.org/
[20] https://lcg-sft.cern.ch/sft/lastreport.cgi
[21] http://goc.grid.sinica.edu.tw/gstat/
[22] http://gridportal.hep.ph.ic.ac.uk/rtm/
[23] http://gridice2.cnaf.infn.it:50080/gridice/site/site.php
[24] http://goc.grid.sinica.edu.tw/gocwiki/FrontPage
[25] http://www.gridpp.ac.uk/deployment/users/
[26] http://www.egee.nesc.ac.uk/
[27] http://indico.cern.ch/conferenceDisplay.py?confId=286
[28] http://goc.grid-support.ac.uk/gridsite/accounting/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Application and Uses of CML within the *e*Minerals project

Toby O. H. White[1], Peter Murray-Rust[2,] Phil A. Couch[3], Rik P. Tyer[2] , Richard P. Bruin[1],

Ilian T. Todorov[3], Dan J. Wilson[4] , Martin T. Dove[1], Kat F. Austen[1] , Steve C. Parker[5]

[1]Department of Earth Sciences, Downing Street, Cambridge. CB2 3EQ
[2]University Chemical Laboratory, Lensfield Road, Cambridge. CB2 1EW
[3]CCLRC Daresbury Laboratory, Daresbury, Warrington. WA4 4AD
[4]Institut für Mineralogie und Kristallographie. J. W. Goethe-Universität, Frankfurt.
[5]Department of Chemistry, University of Bath. BA2 7AY

Abstract

Within the *e*Minerals project we have been making increasing use of CML (the Chemical Markup Language) for the representation of our scientific data. The original motivation was primarily to aid in data interoperability between existing simulation codes, and successful results of CML-mediated inter-code communication are shown. In addition, though, we have discovered several other areas where XML technologies have been invaluable in developing an escientific virtual organization, and benefiting collaboration. These areas are explored, and we show a number of tools which we have constructed. In particular, we demonstrate 1) a general library, FoX for allowing Fortran programs to interact with XML data, 2) a general CML viewing tool, ccViz, and 3) an XPath abstraction layer, AgentX.

1. Introduction

1.1 Introduction to *e*Minerals

The *e*Minerals project is a NERC testbed escience project. Our remit is to study environmentally relevant problems, using molecular-scale modelling techniques, while developing and using escience technologies which directly improve the quality of research we produce.

To this end, the *e*Minerals project encompasses, on the scientific side, researchers from a number of UK universities and research institutions, who represent a broad section of the theoretical and computational environmental science community. These researchers have expertise on a wide variety of modelling codes. Indeed, we have within our team key members of the development teams for several widely used simulation codes (for example, SIESTA has over 1000 users world-wide, and DL_POLY has several thousand.)

On the escience front, therefore, our challenge is to harness this rich expertise, and facilitate collaboration and cross-fertilization between these overlapping areas of science. This paper will show how XML technologies have enabled that, and highlight a number of tools that have resulted from the project.

1.2 Introduction to CML

CML (Chemical Markup Language)[1] was in fact the first example of a full XML language and application. Although initially designed around specifically chemical vocabularies, it has proved very flexible, and more than able to take the additional semantic burden of computational atomic molecular and molecular physics codes.

1.3 *e*Minerals CML background

The eMinerals project has been working with CML for several years now, and we have reported on progress in previous years[2,3]. Our experience has been wholly positive, and CML is playing an increasingly important rôle throughout the project, above and beyond the niches we initially envisaged it filling.

2. CML in *e*Minerals

As mentioned in the introduction, the *e*Minerals project includes scientists from a range of different backgrounds, who work with a wide range of codes. For example, two widely used codes on the project are SIESTA[4], a linear-scaling electronic structure code; and DL_POLY-3[5], a classical molecular dynamics code which uses empirical potentials. In addition, we also use a number of other simulation codes (amongst which are OSSIA[6], RMCProfile [7], METADISE[8]) all written in Fortran.

All of these codes accept and emit chemical, physical, and numerical data, but each uses its own input and output formats. This presents a number of challenges when scientists from different backgrounds, familiar with different codes, have to collaborate:

- when trying to exchange data, format translation and data conversion steps are necessary before different codes can understand the data.
- translation at the human level is necessary, since a scientist familiar with the look and feel of DL_POLY output may not understand SIESTA output, nor know where to look for equivalent data.

Both of these problems can be addressed using XML technologies, and we expand upon this below.

A problem, by no means unique to this project, is that all of our scientific simulation codes are written in Fortran, of varying ages and styles. There are a number of potential approaches to interfacing Fortran and XML; the approach we have adopted is to write an extensive library, in pure Fortran, which exposes XML interfaces in a Fortran idiom. Its design and implementation are briefly explained in section 3.

Having succeeded in making our Fortran codes speak XML, we have found three areas in particular where XML output has been useful. These are briefly explained below, and the tools and methods we have developed are explained in sections 4, 5, and 6.

2.1 Data transfer between codes

When considering the rôle of XML within a project involved in computational science, with multiple simulation codes in use, the temptation is first to think about its use in terms of a common file format which would allow easier data interchange between codes; and indeed that is the perspective from which the *e*Minerals project first approached XML.

The potential uses of the ability to easily share data between simulation codes are manifold. For example, as mentioned previously, we have multiple codes available, and they are capable of doing conceptually similar things, but using different techniques. We might wish to study the same system using both empirical potentials (with DL_POLY) and quantum mechanical DFT (with SIESTA). This would enable us to gain a better appreciation of the different approximations inherent in each method, and better understand the system.

This complementary use of two codes would be made much easier if we could use identical input files for both codes, rather than having to generate multiple representations of the same data. Without any such ability, extra work is required, and there is the potential for errors creeping in as we move between representations.

Furthermore, we might wish to use the output of one code as the input to another. We might wish to extract a small piece of the output of a low accuracy simulation, and study it in much greater depth with our more precise code. Conversely, we might want to take the output of a highly accurate initial calculation, and feed it into a low accuracy code to get more results.

In either of these cases, our workload would be greatly reduced if we could simply pass output directly from one code to another, without concerning ourselves with conversion of representations.

However, the route to this lofty though apparently simple goal of data interoperability is beset by a multitude of complicating factors. We have made much progress towards it, but due to its complications, we have described it last, in section 6.

However, along the way we have discovered several other areas where XML has aided us in our role as an escience testbed project, and these we shall describe first, and expand in sections 4 and 5.

2.2 Data visualization and analysis

One of the major features of XML is the ease with which it may be parsed. Writing an XML parser is by no means trivial, but for almost all languages and toolkits in current use, the work has already been done. Thus, to write a tool which takes input from an XML file, a developer need only interact with an in-memory representation of the XML.

When a computational scientific simulation code is executed, it will produce output in some format which has its origins in a more-or-less humanly readable textual form. Often, though, accretion of output will have rendered it less comprehensible - and in any case a long simulation may well result in hundreds of megabytes of output, which can certainly not be easily browsed by eye.

Output from calculations serves two purposes. First, it enables the scientist to follow the calculation's progress as the job is running, and, once it has finished, to ensure that it has done so in the proper manner, and without errors.

Secondly, it is rare that the scientist is only interested in the fact that the job has finished. Usually, they want to extract further, usually numerical, data from the output, and explore trends and correlations.

These two aims are rarely in accord and this results in output formats having little sensible logical structure. Thus, for some purposes, the scientist will scan the output by eye, while for others, tools must be written to parse the format, and extract and perhaps graphically display relevant data.

XML formats are optimized for ease of computational processing at the expense of human readability. Thus if the simulation output is in XML, scanning by eye becomes infeasible, but processing and visualization tools may be written with much greater ease. If the XML format is common these tools may be of wide applicability. We detail the construction of such tools below, in section 4.

2.3 Data management

In addition to efforts towards data interoperability, and data visualization, there is a third area where we have found XML invaluable, that of data management.

The use of grid computing has brought about an explosion in the volume of data generated by the individual researcher. eScience tools have enabled the generation of multiple jobs to be run, allocation of jobs to available computational resources, recovery of job output, and efficient distributed storage of the resultant data. We have addressed all of these to some extent within our project[9] and we are by no means unique in this.

However, given this large volume of data, categorization is extremely important to facilitate retrieval in both the short term when knowledge on the purpose and context of the data is still to hand and long term, when it may not be. Of equal importance is that when collaborating with geographically disparate colleagues, the data must be comprehensible by both the original investigator who will have some notion of the data's context, and by other investigators who may not.

This is related of course to the perennial problem of metadata, to which there are many approaches. The solution we have come up with in the *e*Minerals project depends heavily on the fact that our data is largely held as XML, which makes it much easier to automatically parse the data to retrieve useful data and metadata by which to index it.

The tools and techniques used are explained further in section 5.

3. Fortran and XML

It was mentioned in the Introduction that we were faced with the problem of somehow interfacing our extensive library of Fortran codes with the XML technologies we wished to take advantage of. There are a number of potential approaches to this issue.

We could write a series of translation scripts, or services, using some XML-aware language, which would convert between XML and whatever existing formats our codes understood. This however requires a multiplication of components, and increases the fragility of our systems.

Alternatively, we could write wrappers for all our codes, again in some XML-aware language, which hid the details of our legacy I/O behind an XML interface. Again though, this is a potentially fragile approach, and requires additional setup of the wrapper wherever we wish to run our codes.

A third potential solution is to reverse the problem, and use Fortran to wrap an existing XML library written in another language, probably C, so that the codes might directly call XML APIs from Fortran, and our workflows ignore the legacy I/O. However, Fortran cross-language compilation is fraught with difficulties, and in addition we would need to ensure that our C library was available and compiled on all platforms where we wished to run.

The solution that we have adopted is to write, from scratch, a full XML I/O library in Fortran, and then allow our Fortran codes to use that.

One of the major advantages of Fortran, and one of the reasons why it is in continued use in the scientific world, is that compilers exist for every platform, and Fortran code is extremely portable across the 9 or 10 compilers, and 7 or 8 hardware platforms, which are commonly available. There is no XML-aware language which is as portable, and cross-language compilation over that number of potential systems is a painful process. So, although writing the whole library from scratch in Fortran does involve more work than leveraging an existing solution, it achieves maximum portability, and means we are not restricted at all upon where our XML-aware codes may run.

The library we have written is called FoX (Fortran XML)[10] and an earlier version (named xmlf90) was described in [2]. In its current incarnation, it consists of four modules, which may be used together or independently. Two of them provide APIs for input APIs, and two for output.

On the input side, the modules provide APIs for event-driven processing, SAX 2[11], and for DOM Core level 2[12]. On the output side, there is a general-purpose XML writing library; built on top of which is a highly abstracted CML output layer. It is written entirely in standard Fortran 95.

The SAX and DOM portions of the library contain all calls provided by those APIs, modified slightly for the peculiarities of Fortran. An overview of their initial design and capabilities may be seen in [2] although the current version of FoX has significantly advanced, not least in now having full XML Namespaces[13] support. The two output modules, wxml and wcml are described here.

3.1 wxml

Clearly XML can be output simply by a series of print statements, in any language. However, XML is a tightly constrained language, and simple print statements are firstly very prone to errors, and secondly, make correct nesting of XML across a document impossible for anything but the most simple of applications. The requirement for good XML output libraries has long been recognized.

However, for most languages, simple XML output libraries rarely exist, or tend to be second-class citizens. More usually, XML output tends to be a simple addendum to a DOM library. If so, a method is provided which serializes the in-memory DOM. Thus as long as your data structures are held as a tree in memory, XML output is trivial.

Indeed, FoX supplies such a method with its DOM implementation. For applications that can easily be written around a tree-like data structure, this is fine. However, for nearly all existing Fortran applications, this is of no use whatsoever; Fortran is not a language designed with tree-like data-structures in mind, and in any case most Fortran developers are unfamiliar with anything more complicated than arrays. Forcing all data in a simulation code to be held within a DOM-like model would be entirely unnatural.

Furthermore, simulation codes often produce extremely large quantities of data, and may do so over extended periods of time. It would be foolish to keep this data all in memory for the entirety of the simulation. Not only would it be a vast waste of

memory, but since one would only be able serialize once the run is over and all the data complete, if the run were interrupted, all data would be lost. In addition, it is very important that one be able to keep track of the simulation as it progresses by observing its output, which requires progressive output, rather than all-in-one serialization.

Much of this could be avoided, of course, by occasional serialization of a changing DOM tree; or even by temporary conversion of the large Fortran arrays to a DOM and then step-by-step output; or, indeed, by the method we use, which is direct XML serialization of the Fortran data.

Thus, the FoX XML output layer (termed *wxml* here) is a collection of subroutines which generate XML output - tags, attributes and text - from Fortran data. The library is extensive, and allows the output of all structures described by the XML 1.1 & XML Namespaces specifications[13,14], although for most purposes only a few are necessary. XML output is obtained by successive calls to functions named **`xmlNewElement`**, **`xmlAddAttribute`**, and **`xmlEndElement`**. Furthermore, there are a series of additional function calls to directly translate native Fortran data structures to appropriate collections of XML structures. State is kept by the library to ensure well-formedness throughout the document.

Thus *wxml* enables rapid and easy production of well-formed XML documents, providing the user has a good grasp of the XML they want to produce.

3.2 wcml

However, the impetus for the creation of this library was the desire to graft XML output onto existing codes; specifically CML output. *wxml* is insufficient for this for a number of reasons:

- the developers of traditional Fortran simulation codes are, by and large, ignorant of XML, and entirely content to stay that way.
- when adapting a code to output XML, it is important that any changes made not obscure the natural flow of the code.
- it is desirable to write specialized functions to output CML elements, and common groupings thereof, directly (if for no other reason than to avoid misspellings of element and attribute names in source code.)
- further, it is useful to maintain a “house style” for the CML.

Thus, for example, consider a code that wishes to output the coordinates of a molecular configuration. This is represented in CML by a 3-level hierarchy of various tags, with around 20 different optional attributes, and 4 different ways of specifying the coordinates themselves. The average simulation code developer does not wish to concern themselves with the minutiae of these details; they certainly do not wish to clutter up the code with hundreds of XML output statements.

In fact, were this output to be encoded in the source as a lengthy series of loops over *wxml* calls, it is almost certain that even if it were written correctly initially, as the code continued to be developed by XML-unaware developers, it would eventually break.

Therefore, FoX provides an interface where these details are hidden from view:

```
call cmlAddMolecule(xf=xmlFile,
      coords=array_of_coords,
      elems=array_of_elements)
```

which outputs the following chunk of CML:

```
<molecule>
  <atomArray>
    <atom xyz3=”0.1 0.1 0.1”
      elementType=”H”/>
    ....
  </atomArray>
</molecule>
```

Similar interfaces are provided for all portions of CML commonly used by simulation codes.

Such calls do not interfere with the flow of the code, and it is immediately obvious what their purpose is. Further, with such calls, a developer with only a rudimentary knowledge of XML/CML can easily add CML output to their code.

This *wcml* interface is now used in most of the Fortran simulation codes within *e*Minerals, including the current public releases of SIESTA 2.0 and DLPOLY 3, and in addition is in the version of CASTEP[15] being developed by MaterialsGrid[16]. FoX is freely available, and BSD licensed to allow its inclusion into any products - we encourage its use.

4. Data visualization

Although reformatting simulation output in CML allows for much richer handling of the data, it has the aforementioned disadvantage that the raw output is then much less readable to the human eye than raw textual output.

This led to the desire for an tool which would translate the CML into something more comprehensible. At its most basic level, this could simply strip away much of the angle-bracketed-verbiage associated with XML. However, since a translation needs to be done anyway, it is then not much more trouble to ensure that the translated output is visually much richer.

Such tools have been built before on non-XML bases, with adapters for the output of different codes, but what we hoped to do here was, through the use of XML, avoid the necessity for the viewing tool to be adapted for new codes. In addition, it was strongly desired that the viewing, as far as possible, require no extra software installation from the user, in order that data could be viewed by colleagues and collaborators as easily as possible.

Since CML is an XML language, and translations between XML languages are easily done; and XHTML is an XML language; and furthermore, these days the web-browser is an application platform in its own right, which is present on everyone's desktop

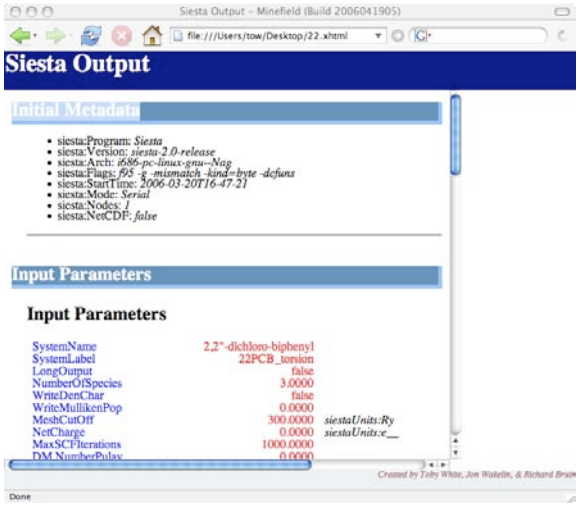


Figure 1: marked up metadata and parameters

already; we followed the route of transforming our CML into XHTML.

The browser application platform offers rich presentation of textual data, and rich presentation of graphical data, which can also be in an XML format, such as SVG (Scalable Vector Graphics). Furthermore, there is the possibility of interactivity, through the use of Javascript (JS) to manipulate the displayed XHTML/SVG. Finally, it affords the opportunity to to embed Java applets, for interaction on a level beyond what JS+XML can do; especially in this case through the use of Jmol[17], which is a pre-existing, well-featured molecular visualization platform.

Therefore, we wished to transform our CML into XHTML. This could be accomplished by many methods, but, XSLT was the obvious choice, since it is explicitly designed to convert between XML languages. Further, modern web-browsers have limited, but increasing, support for performing XSLT transformations themselves. This held out the possibility that we could rely on the browser to perform the transformation as well as the rendering of the output. We would then be able simply to point the web browser at the raw CML, and conversion would occur automatically. and in addition, is interpretable by the browser itself; thus it should be possible to view the CML file directly in the browser and have the transformation take place as the document is rendered.

The result of this process was a set of XSLT transforms which we term **ccViz** (computational chemistry Visualizer). Browser XSLT capabilities are sadly not yet at the stage where they can perform the XSLT, but nevertheless **ccViz** has proved invaluable.

The XSLT transformation starts with simple mark-up of quantities with their names and units extracted and placed together. Thus instead of looking through the CML to find the total energy, the name “Total Energy” is marked up in colour, and its value shown, with units attached. This is shown in figure 1. Although the resultant page is nicer to look at, and immediately more readable, this transformation is very straightforward. However, two particular further aspects of the transform deserve attention.

4.1 SVG graphs

Firstly, of note is the production of SVG graphs from the CML data. For a simulation output, it is valuable to see the variation of some quantities as the simulation progresses; of temperature, or total energy, for example. This can be done with a table of numbers, and indeed traditionally has been - the simulation scientist grows used to casting their eye down a list of numbers to gauge variation when viewing text output files, in the absence of a better solution. However, a line graph is much easier to grasp. SVG is ideally suited for rendering such graphs, as a 2D vector language. A generalized line-graph drawing XSLT library was written, which, when fed a series of numbers, will calculate offsets & draw a well-proportioned graph, with appropriate labels and units. The visualization transform can then pull out any relevant graphable data (which may be determined solely from the structure of the document, independent of the simulation code used), and produces graphs, which are then embedded inline into the XHTML document.

The transform is performed entirely within XSLT, without recourse to another language, which makes it embeddable in the browser engine. An example is shown in figure 2. The plotting engine is known as Pelote and is freely available for use.

Thus, a mixed-namespace XHTML/SVG document is produced, and on viewing the output file, it is immediately easy to see the progress of the simulation by eye. This is of great importance, augmenting productivity by increasing the ease with which the researcher may monitor the progress of their simulations, particularly in a grid-enabled environment, with many concurrent jobs running.

4.2 Jmol viewing

Since the outputs of all of our codes concern molecular configurations, it would be extremely useful to be able to see and manipulate the 3D molecular structures generated by the simulation. This is a task it is impossible to perform by eye from a listing of coordinates. However, there is no 3D XML language which is sufficiently widely

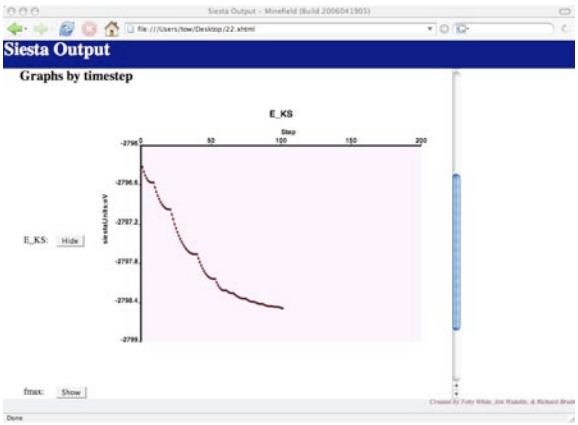


Figure 2: automatically generated SVG graph

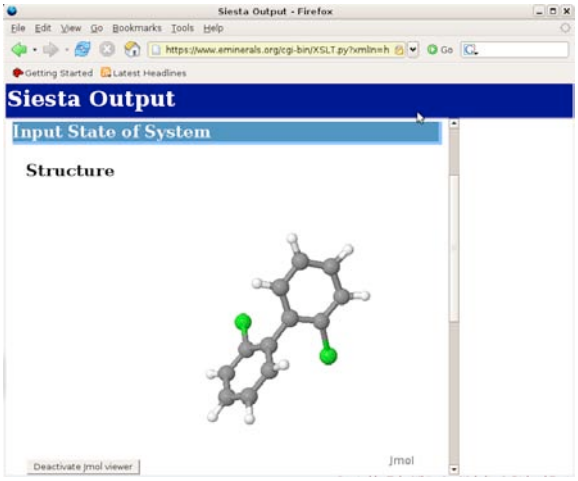


Figure 3: Interactive Jmol applet embedded in webpage.

implemented in browsers to make this doable in the fashion we managed for 2D graphs.

Fortunately, the well-established tool, Jmol[17], which knows how to read CML files, is primarily designed to act as an applet embedded in web-pages. We may therefore use our XSLT transform to embed instances of Jmol into the viewable output. However, Jmol accepts input only as files or strings, so we needed to find a way to pass individual different molecular configurations from our monolithic CML document to different individual Jmol applets embedded in the transformed webpage. We overcame this problem by producing multiple-namespace XML documents: the output contains not only XHTML and SVG, but also chunks of CML, containing the different configurations of interest. Since the browser is unaware of CML, it makes no attempt to render this data. We then took advantage of Java/Javascript interaction to enable its use.

There is an API for controlling the browser's JS DOM from a Java applet, called LiveConnect. We adapted the Jmol source code so that when passed a string which is the id of a CML tag in the embedding document, Jmol will traverse the in-memory browser DOM to find the relevant node, and extract data from the CML contained therein. This enables us to have a single XHTML document directly containing all the CML data, which knows how to interpret itself. This addition to Jmol has been included in the main release, and is available from version 10.2. An example is illustrated in figure 3.

4.3 Final document

Thus, we constructed an XSLT transform which, when given a CML document, will output an XHTML/SVG/CML document which is viewable in a web-browser; in which all relevant input & output data is marked-up and displayed; all time-varying data is graphed, and all molecular configurations can be viewed in three dimensions and manipulated by the user.

Very conveniently, the XSLT transform knows nothing about what these quantities are, nor from what code they originate. So, we may add new quantities to the simulation output, and they are automatically picked up and displayed in the browser, and graphed if appropriate.

Furthermore, it is important to note that any other code which produces CML output is viewable in the same way. This is useful because one of the biggest barriers to be overcome in sharing data between scientists from different backgrounds is a lack of familiarity with each other's toolsets. However, with ccViz, we can share data between DL_POLY and SIESTA users and view them in exactly the same way.

This transferability applies not only to this visualization tool, but to any analysis tools built on CML output - a tool which analyses molecular configurations, or which performs statistical analyses on timestep-averaged data, for example, need be written only once, since the raw information it needs may be extracted from the output file the same way for every CML-enabled code. Previously, such a tool would rely on the output format of one particular simulation code.

5. Data management

As mentioned above, working within a grid-enabled environment, it is easy to generate large quantities of data. The problem then arises of how to manage this data. Within the *e*Minerals project we store our data using the San Diego Supercomputing Centre's SRB [18], though there are many other solutions to achieve similar aims of distributed data storage.

However, the major problem encountered is not where to store the data, but how to index and retrieve it; for both the originator of the data and their collaborators. The *e*Minerals project faces a particular challenge in this regard, since we have a remarkably wide variety of data being generated. XML helps in this task in two ways. Firstly, having a common approach to output formats gives the advantages explained in the previous section. The second, larger, issue is the general problem of metadata handling, and approaches have been attempted with varying degrees of success by many people. The *e*Minerals approach is explained in great detail in [19]. However, we shall discuss it briefly here, with particular reference to the ways in which XML has helped us solve the problem.

There are three sorts of metadata which we have found it useful to associate with each simulation run:

- metadata associated with the compiled simulation code - its name, its version, any compiled-in options, *etc*. Typically there will be 10 or so such items.
- metadata associated with the set up of the simulation; input parameters, in other words. These will vary according to the code - some codes may use the temperature and pressure of the model system, some may record what methods were used to run the model, *etc.* Typically there will be 50 or so such parameters
- Job-specific metadata. Since the purpose of metadata is in order to index the job for later, easy, retrieval, sometimes it is appropriate to attach extracted output as metadata. For example, if a

number of jobs are being run, with the final model system energy being of particular interest, it is useful to attach the value of this quantity as metadata to the stored files in order that the jobs may be later examined, indexed by this value.

In each of these cases, we find our life made much easier due to the fact that our files are in an XML format.

For the first two types of metadata, we use the fact that CML offers **<metadata>** and **<parameter>** elements, which have a (simplified) structure like the following:

```
<metadataList>
  <metadata name="Program"
content="DL_POLY">
  ...
</metadataList>
```

We may therefore extract all such elements and store them as metadata values in our metadata database.

The third type of metadata obviously requires specific instructions for the job at hand. However, because the output files are in XML format, we may easily point to locations within the file such that tools can automatically extract relevant data. This can be done easily using an XPath expression to point at the final energy, for example (although in fact we do not use XPath directly - we use AgentX[20], as detailed in the next section.)

6. Data transfer

Finally, we have also succeeded in using XML to work towards code interoperability in the fashion originally foreseen.

The idea of a common data format is attractive, and as explained in section 2, would be of enormous value. It has, however, proved elusive so far, for a number of reasons, the discussion of which is beyond the scope of this paper. Nevertheless, by passing around small chunks of well-formatted, well-specified data, and agreeing on a common dialect with a very small vocabulary, we are able to gain much in interoperability.

Much of the progress we have made in this area is due to AgentX, an XML abstraction tool we have built to help us in this task.

6.1 AgentX

AgentX is a tool originating primarily from CCLRC Daresbury, but in the development of which *e*Minerals has played an important part. A previous version was described in [20].

At its most basic, it may be understood as, firstly, a method of abstracting complicated XPath expressions, and secondly as a method of abstracting access to common data which may expressed differing representations by various XML formats.

For example, the three-dimensional location of an atom within a molecule is expressed in CML, as shown in section 3.2 above, with nested **<molecule>**, **<atomArray>**, and **<atom>** elements. AgentX provides an interface by which one can access that data in terms of the data represented rather than the details of that representation, as illustrated by the following, simplified, series of pseudocode API calls.

```
axSelect("Molecule")
numAtoms = axSelect("Atom")
for i in range(numAtoms):
   axSelect('xCoordinate')
   x = axValue()
   axSelect('yCoordinate')
   y = axValue()
   axSelect('zCoordinate')
   z = axValue()
```

Internally, this is implemented by AgentX having access to three documents - the CML source, an OWL[21] ontology, and an RDF[22] mapping files.

"Molecule" is a concept defined in the OWL ontology; and the RDF provides a mapping between that concept and an XPath/Xpointer expression, which evaluates to the location of a **<molecule>** tag. Furthermore, the ontology indicates that "Molecule"s may contain "Atom"s, which may have properties "xCoordinate", "yCoordinate", and "zCoordinate". The RDF provides mappings between each of these concepts, and XPath expressions which point to locations in the CML.

Thus, presented with a CML file containing a **<molecule>**, it may be queried with AgentX to retrieve the atomic coordinates by a simple series of API calls, without the need to understand the syntax of the CML file.

More powerfully, though, it is possible to specify multiple mappings for one concept. That is, we may say that a "Molecule" may be found in multiple potential locations.

The normal CML format for specifying a molecule is quite verbose, albeit clear. For DL_POLY we needed to represent tens of thousands of atoms efficiently, so we used CML's **<matrix>** element to provide a compressed format, at the cost of losing the contextual information provided by the CML itself.

However, by simply providing a new set of mappings to AgentX, we could inform it that "Molecule"s could be found in this new location in a CML file, so all AgentX-enabled tools could immediately retrieve molecular and atomic data without any need for knowledge of the underlying format change.

AgentX is implemented as an application on top of libxml2. It is implemented primarily in C, with wrappers to provide APIs for Perl, Python and Fortran.

Concepts and mappings are provided for most of the data that are in common use throughout the project, but it is easy to add private concepts or further mappings where the existing ones are insufficient.

6.2 Data interchange between codes

We have incorporated AgentX into the CML-aware version of DL_POLY. This has enabled us to use CML-encoded atomic configurations as a primary storage format for grid-enabled studies. Details of studies performed with this CML-aware DL_POLY are in [23].

Furthermore, the output of any of our CML-emitting codes may now be directly used as input to DL_POLY, so we can perform direct comparisons of SIESTA and DL_POLY results.

AgentX has also been linked into a number of other codes., including AtomEye[24]; and the CCP1 GUI[25]. It is also used as the metadata extraction tool in the scheme described in section 5.

Thus we are now successfully using CML as a real data interchange format between existing codes that have been adapted to work within an XML environment.

7. Summary

Within the *e*Minerals project, we have made wide, and increasing, use of XML technologies, especially with CML.

While working towards the goal of data interoperability between environmentally-relevant simulation codes, we have found several additional areas where XML has been of particular use, and have developed a number of tools which leverage the power of XML technologies to enable better collaborative research and eScience. These include:

- FoX, a pure Fortran library for general XML, and particular CML, handling.
- Pelote, an XSLT library for generating SVG graphs from
- ccViz, an XHTML-based CML viewing tool.
- AgentX, an XML data abstraction layer.

All of our tools are liberally licensed, and are freely available from http://www.eminerals.org/tools

Finally, we have successfully developed methods for true interchange of XML data between simulation codes.

Acknowledgements

We are grateful for funding from NERC (grant reference numbers NER/T/S/2001/00855, NE/C515698/1 and NE/C515704/1).

References

[1] Murray-Rust, P and Rzepa, H. S., *"Chemical Markup Language and XML Part I. Basic principles"*, J. Chem. Inf. Comp. Sci., **39**, 928 (1999);
Murray-Rust, P. and Rzepa, H.S., *"Chemical Markup, XML and the World Wide Web. Part II: Information Objects and the CMLDOM"*, J. Chem. Inf. Comp. Sci., **41**, 1113 (2001).

[2] Garcia, A., Murray-Rust, P. and Wakelin, J. *"The use of CML in Computational Chemistry and Physics Programs"*, All Hands Meeting, Nottingham, 1111. (2004)

[3] White, T.O.H. *et al. "eScience methods for the combinatorial chemistry problem of adsorption of pollutant organic molecules on mineral surfaces"*, All Hands Meeting, Nottingham, 773 (2005)

[4] Soler, J. M. *et al, " The Siesta method for ab initio order-N materials simulation"*, J. Phys.: Condens. Matter, **14**, 2745 (2002).

[5] Todorov, I., and Smith, W., Phil. Trans. R. Soc. Lond. A, **362**, 1835. (2004)

[6] Warren. M.C. *et al., "Monte Carlo methods for the study of cation ordering in minerals."* , Mineralogical Magazine **65**, (2001); also http://www.esc.cam.ac.uk/ossia/

[7] M. G. Tucker, M. T. Dove, and D. A. Keen, J. Appl. Crystallogr. 34, 630 (2001)

[8] Watson, G.W. *et al., "Atomistic simulation of dislocations, surfaces and interfaces in MgO"* J. Chem. Soc. Faraday Trans., **92**(3), 433 (1996); also http://www.bath.ac.uk/~chsscp/group/programs/programs.html

[9] Bruin, R.P., *et al., "Job submission to grid computing environments"*, All Hands Meeting, Nottingham (2006), and references therein.

[10] This is not the only Fortran XML library in existence, - see also http://nn-online.org/code/xml/, http://sourceforge.net/projects/xml-fortran/, http://sourceforge.net/projects/libxml2f90 - but it is the most fully featured. Its output support surpasses others and it is certainly the only one with CML support.

[11] http://www.saxproject.org

[12] http://www.w3.org/DOM/

[13] Bray, T. *et al., "Namespaces in XML 1.1"*, W3C Recommendation, 4 February 2004

[14] Bray, T. *et al., "Extensible Markup Language (XML) 1.1"* ,W3C Recommendation, 4 February 2004

[15] Segall, M.D. *et al.*, J. Phys.: Cond. Matt. **14**(11) pp.2717-2743 (2002)

[16] http://www.materialsgrid.org

[17] http://jmol.sourceforge.net

[18] http://www.sdsc.edu/srb

[19] Tyer, R.P. *et al., "Automatic metadata capture and grid computing"*, All Hands Meeting, Nottingham (2006) - in press.

[20] Couch, P.A. *et al.,"Towards Data Integration for Computational Chemistry"* All Hands Meeting, Nottingham 426 (2005)

[21] http://www.w3.org/TR/owl-features/

[22] http://www.w3.org/RDF/

[23]Dove, M.T. *et al., "Anatomy of a grid-enabled molecular simulation study: the compressibility of amorphous silica"*, All Hands Meeting, Nottingham (2006)

[24] Li, J., *"Atomeye: an efficient atomistic configuration viewer"*, Modelling Simul. Mater. Sci. Eng., **173** (2003)

[25] http://www.cse.clrc.ac.uk/qcg/ccp1gui/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

An Architecture for Language Processing for Scientific Texts

Ann Copestake[1], Peter Corbett[2], Peter Murray-Rust[2], CJ Rupp[1], Advaith Siddharthan[1], Simone Teufel[1], Ben Waldron[1]

[1] Computer Laboratory, University of Cambridge
[2] Unilever Centre for Molecular Informatics, University of Cambridge

Abstract

We describe the architecture for language processing adopted on the eScience project 'Extracting the Science from Scientific Publications' (nicknamed SciBorg). In this approach, papers from different sources are first processed to give a common XML format (SciXML). Language processing modules operate on the SciXML in an architecture that allows for (partially) parallel deep and shallow processing and for a flexible combination of domain-independent and domain-dependent techniques. Robust Minimal Recursion Semantics (RMRS) acts both as a language for representing the output of processing and as an integration language for combining different modules. Language processing produces RMRS markup represented as standoff annotation on the original SciXML. Information extraction (IE) of various types is defined as operating on RMRSs. Rhetorical analysis of the texts also partially depends on IE-like patterns and supports novel methods of information access.

1 Introduction

The eScience project 'Extracting the Science from Scientific Publications' (nicknamed SciBorg, `www.sciborg.org.uk`) aims to build dynamic, flexible and expandable natural language processing (NLP) infrastructure which will support applications in eScience. We hope to show that autonomous, adaptive methods, based on NLP techniques, can be used to mine the primary literature and other text to build an evolving knowledge base for eScience. Overall, the goals of SciBorg are:

1. To develop a markup language for natural language which will act as a platform for extraction of information.
2. To develop Information Extraction (IE) technology and core ontologies for use by publishers, researchers, readers, vendors, and regulatory organisations.
3. To model scientific argumentation and citation purpose in order to support novel modes of information access.
4. To demonstrate the applicability of this infrastructure in a real-world eScience environment.

The SciBorg project started in October 2005 and is a collaboration between the Computer Laboratory, the Unilever Centre for Molecular Informatics and the Cambridge eScience Centre, with support from the Royal Society of Chemistry, Nature Publishing Group and the International Union of Crystallography. We are concentrating on Chemistry texts in particular, but we aim to develop techniques which are largely domain-independent with clear interfaces to domain-dependent processing. For instance, we are using some of the same tools as the Flyslip project (Hollingsworth et al., 2005; Vlachos and Gasperin, 2006), which concerns extraction of functional genomic information to aid FlyBase curation. We are also collaborating with the Citation Relations and Argumentative Zoning (CitRAZ) project, especially on the discourse processing aspects of the project.

The goal of this paper is to introduce and motivate the architecture which we have adopted for language processing within SciBorg and to describe some of the progress so far on implementing that architecture and the various language processing modules it comprises.

Characteristic features of our general approach to language processing are:

1. We are integrating, adapting and further developing general tools for language processing. We intend to avoid domain-specific solutions wherever possible, even if this leads to lower performance in the short term. The primary goal of the project is to improve the language technology.
2. We are incorporating relatively deep syntactic and compositional semantic processing. By 'deep',

we mean systems which use very precise and detailed grammars of natural languages to analyse and generate, especially approaches based on current linguistic theory (see §5).
3. We are developing a semantically-based representation language Robust Minimal Recursion Semantics (RMRS:Copestake (2003)) for integration of all levels of language processing and to provide a standardised output representation. RMRS is an application-independent representation which captures the information that comes from the syntax and morphology of natural language while having a sound logical basis compatible with Semantic Web standards. Unlike previous approaches to semantic representation, underspecified RMRSs can be built by shallow language processing, even part-of-speech (POS) taggers. See §4.
4. Integration with XML is built in to the architecture, as discussed in §3. We view language processing as providing standoff annotation expressed in RMRS-XML, with deeper language processing producing more detailed annotation.
5. As far as possible, all technology we develop is Open Source. Much of it will be developed in close collaboration with other groups.
Our overall aim is to produce an architecture that allows robust language processing even though it incorporates relatively non-robust methods, including deep parsing using hand-written grammars and lexicons. The architecture we have developed is not pipelined, since shallow processing can operate in parallel to deeper processing modules as well as providing input to them. We do not have space in this paper for detailed comparison with previous approaches, but note that this work draws on results from the Deep Thought project: Uszkoreit (2002) discusses the role of deep processing in detail.
In the next section, we outline the application tasks we intend to address in SciBorg. This is followed by a description of the overall architecture and the RMRS language. §5 and §6 provide a little more detail about the modules in the architecture, concentrating on architecture and integration details. §7 describes our approach to discourse analysis.

2 Application tasks

In this section, we outline three tasks that we intend to address in the project. As we will explain below, these tasks all depend on matching patterns specified in terms of RMRS. They can all be considered as forms of information extraction (IE), although we use that term very broadly. Most existing IE technology is based on relatively shallow processing of texts to directly instantiate domain-specific templates or databases. However, for each new type of information, a hand-crafted system or an extensive manually-created training corpus is required. In contrast, we propose a layered architecture using an approach to IE that takes the RMRS markup, rather than text, as a starting point. Again, this follows Deep Thought, but only a limited investigation of the approach was attempted there.

Chemistry IE We are interested in extraction of specific types of chemistry knowledge from texts in order to build a database of key concepts fully automatically. For example, the following two sentences are typical of the part of an organic synthesis paper that describes experimental methods:

> The reaction mixture was warmed to rt, whereat it was stirred overnight. The resultant mixture was kept at 0C for 0.5 h and then allowed to warm to rt over 1 h.

Organic synthesis is a sufficiently constrained domain that we expect to be able to develop a formal language which can capture the steps in procedures. We will then extract 'recipes' from papers and represent the content in this language. A particular challenge is extraction of temporal information to support the representation of the synthesis steps.

Ontology construction The second task involves semi-automatically extending ontologies of chemical concepts, expressed in OWL. Although some ontologies exist for chemistry and systematic chemical names also give some types of ontological information (see §6.2), resources are still relatively limited, so automatic extraction of ontological information is important. Our proposed approach to this is along the lines pioneered by Hearst (1992) and refined by other authors. For instance, from:

> ...the concise synthesis of naturally occurring alkaloids and other complex polycyclic azacycles.

we could derive an expression that conveyed the information an 'alkaloid IS-A azacycle'.

```
<owl:Class rdf:ID="Alkaloid">
<rdfs:subClassOf
       rdf:resource="#Azacycle"\>
```

Ontologies considerably increase the flexibility of the chemistry IE, for instance by allowing for matches on generic terms for groups of chemicals (cf., the GATE project (`gate.ac.uk`) 'Ontology Based Information Extraction/OBIE').

Research markup The third application is geared towards humans browsing papers: the aim is to help them quickly see the most salient points and the interconnections between papers. The theoretical background to this is discussed in more detail in

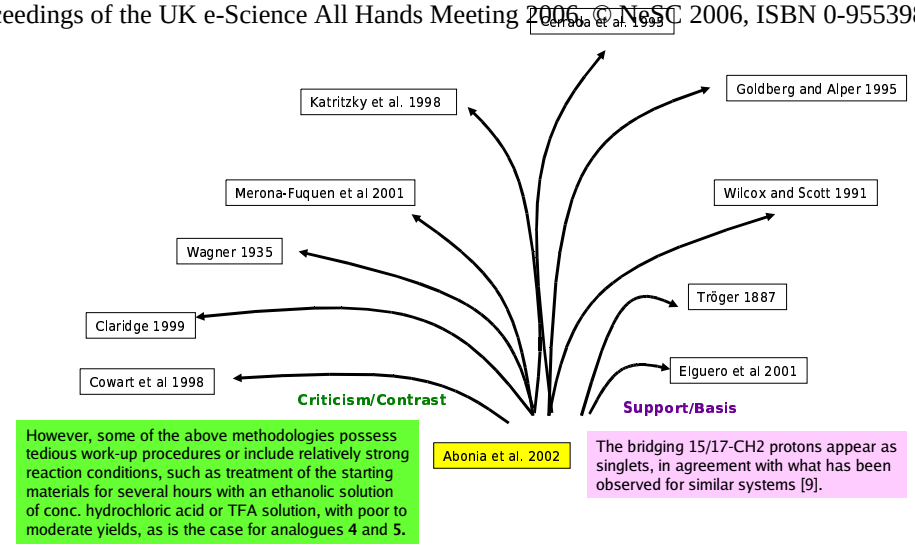


Figure 1: A rhetorical citation map

§7, but here we just illustrate its application to citation. Our aim is to provide a tool which will enable the reader to interpret the rhetorical status of a citation at a glance, as illustrated in Fig. 1. The example paper (Abonia et al, 2002) cites the other papers shown, but in different ways: Cowart et al (1998) is a paper with which it contrasts, while Elguero et al (2001) is cited in support of an observation. We also aim to extract and display the most important textual sentence about each citation, as illustrated in the figure.

The Chemistry Researchers' amanuensis By the end of the project, we will combine these three types of IE task in a research prototype of a 'chemistry researchers' amanuensis' which will allow us to investigate the utility of the technology in improving the way that chemists extract information from the literature. Our publisher partners are supplying us with large numbers of papers which will form a database with which to experiment with searches. The searches could combine the types of IE discussed above: for instance, a search might be specified so that the search term had to match the goal of a paper.

3 Overall architecture

Figure 2 is intended give an idea of the overall architecture we are adopting for language processing. The approach depends on the initial conversion of papers to a standard version of XML, SciXML, which is in use not just on SciBorg but on the FlySlip and CITRAZ projects as well (Rupp et al., 2006). For SciBorg, we have the XML source of papers, while for FlySlip and CitRAZ, the source is pdf. Following this conversion, we invoke the domain-specific and domain-independent language processing modules on the text. In all cases, the language processing adds standoff annotation to the SciXML base. Standoff annotation uses SAF (Waldron and Copestake, 2006; Waldron et al., 2006). SAF allows ambiguity to be represented using a lattice, thus allowing the efficient representation of multiple results from modules.

Language processing depends on the domain-dependent OSCAR-3 module (see §6) to recognise compound names and to markup data regions in the texts. The three sentence level parsing modules shown here (described in more detail in §5) are the RASP POS tagger, the RASP parser and the ERG/PET deep parser. The language processing modules are not simply pipelined, since we rely on parallel deep and shallow processing for robustness. Apart from the sentence splitter and tokenisers, all modules shown output RMRS. Output from shallower processing can be used by deeper processors, but might also be treated as contributing to the final output RMRS, with RMRSs for particular stretches of text being selected/combined by the RMRS merge module. The arrows in the figure indicate the data flows which we are currently using but others will be added: in later versions of the system the ERG/PET processor will be invoked specifically on subparts of the text identified by the shallower processing. The deep parser will also be able to use shallow processing results for phrases within selected areas where there are too many out-of-vocabulary items for deep parsing to give good results.

Processing which applies 'above' the sentence level, such as anaphora resolution and (non-syntactically marked) word sense disambiguation (WSD), will operate on the merged RMRS, enriching it further. However, note that the SciXML markup is also accessible to these modules, allowing for sensitivity to paragraph breaks and section boundaries, for instance. The application tasks are all defined to operate on RMRSs.

This architecture is intended to allow the benefits of deep processing to be realised where possible but with shallow processing outputs being used as nec-

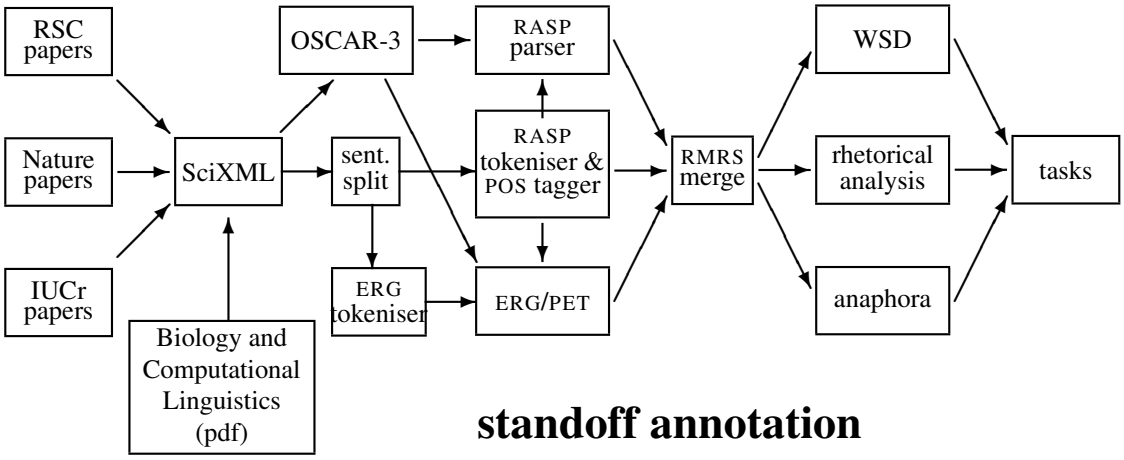


Figure 2: Overall architecture of the SciBorg system

essary for robustness. The aim is to always have an RMRS for a processed piece of text. The intent is that application tasks be defined to operate on RMRSs in a way that is independent of the component that produced the RMRS. Other modules could be added as long as they can be modified to produce RMRSs: we may, for instance, investigate the use of a chunker. The architecture allows the incorporation of modules which have been independently developed without having to adjust the interfaces between individual pairs of modules.

The parallel architecture is a development of approaches investigated in VerbMobil (Ruland et al., 1998; Rupp et al., 2000) and Deep Thought (Callmeier et al., 2004). It is intended to be more flexible than either of these approaches, although it has resemblances to the DeepThought Heart of Gold machinery and we are using some of the software developed for that. The QUETAL project (Frank et al., 2006) is also using RMRS. We have discovered that a strictly pipelined architecture is unworkable when attempting to combine modules that have been developed independently. For instance, different parsing technology makes very different assumptions about tokenisation, depending in particular on the treatment of punctutation. Furthermore, developers of resources change their assumptions over time, so attempting to make tokenisation consistent leads to a brittle system. Our non-pipelined, standoff annotation approach supports differences in tokenisation. It is generic enough to allow us to investigate a range of different strategies for combining deep and shallow processing.

4 RMRS

RMRS is an extension of the Minimal Recursion Semantics (MRS: Copestake et al. (2005)) approach which is well established in deep processing in NLP. MRS is compatible with RMRS but RMRS can also be used with shallow processing techniques, such as part-of-speech tagging, noun phrase chunking and stochastic parsers which operate without detailed lexicons. Shallow processing has the advantage of being more robust and faster, but is less precise: RMRS output from the shallower systems is less fully specified than the output from the deeper systems, but in principle fully compatible. In circumstances where deep parsing can be successfully applied, a detailed RMRS can be produced, but when resource limitations (in processing capability or lexicon availability, for instance) preclude this, the system can back-off to RMRSs produced by shallower analysers. Different analysers can be flexibly combined: for instance shallow processing can be used as a preprocessor for deep analysis to provide structures for unknown words, to limit the search space or to identify regions of the text which are of particular interest. Conversely, RMRS structures from deep analysis can be further instantiated by anaphora resolution, word sense disambiguation and other techniques. Thus RMRS is used as the common integration language to enable flexible combinations of resources.

Example RMRS output for a POS tagger and a deep parser for the sentence *the mixture was allowed to warm* is shown below:[1]

```
Deep processing          POS tagger
h6:_the_q(x3)            h1:_the_q(x2)
  RSTR(h6,h8)
  BODY(h6,h7)
h9:_mixture_n(x3)        h3:_mixture_n(x4)
  ARG1(h9,u10)
h11:_allow_v_1(e2)       h5:_allow_v (e6)
  ARG1(h11,u12)
  ARG2(h11,x3)
  ARG3(h11,h13)
```

[1]For space reasons, we have shown a rendered format, rather than RMRS-XML, and have omitted much information including tense and number.

```
  qeq(h13,h17)
h17:_warm_v(e18)        h7:_warm_v(e8)
  ARG1(h17,x3)
```

RMRSs consist of 'flat' structures where the information is factorised into minimal units. This facilitates processing and is key to the approach to underspecification. Predicates correspond to morphological stems, annotated with 'v', 'n' etc to give a coarse-grained indication of sense. These names can be constructed automatically on the basis of POS-tagged text. The POS tagged text can thus share the same lexicalised predicates as the deep parser output (_mixture_n, _allow_v, _warm_v, _the_q), although the deep parser can make more fine-grained sense distinctions (allow_v_1) and may insert predications that arise from particular grammatical constructions, such as compound nouns (not shown here).

The POS tagger has no relational information (indicated in the deep output by ARG1 etc). In the deep parser, predicates are specified in the lexicon and are of fixed arity. Uninstantiated relational positions in the deep output are indicated by 'u's, 'e's are eventualities, and 'x's other entities. The qeq condition in the deep output is a partial scope constraint which relates 'h' labels.

RMRS has been designed to be suitable for natural language representation and as such has to be very expressive while at the same time allowing for underspecification. Formally, RMRSs (like MRSs) are partial descriptions which correspond to a set of logical forms in a higher-order base language. RMRS itself is a restricted first order language: scope relationships are reified (via the 'h' labels) and natural language quantifiers, such as *every* and *most*, correspond to predicates, though these in turn correspond to generalised quantifiers in the base language. Inference in the base language will not, in general, be tractable, but some inferences can be directly expressed using RMRS without resolving to the base language. RMRSs can be linked to ontologies, so that the notion of underspecification of an RMRS reflects the hierarchical ontological relationship.

5 Domain-independent sentence processing modules

Apart from OSCAR-3 (see next section), the modules shown in Figure 2 are essentially domain-independent. Not all modules are shown explicitly. Parsing depends on the text being initially split into chunks (typically, but not necessarily, sentences). Domain-specific processing is required to identify some regions as unsuitable for passing to the parsers (e.g., data sections).

The three modules shown in Figure 2 have all been developed previously and used in a variety of applications. The RASP part of speech tagger (Briscoe and Carroll, 2002) statistically determines tags for individual tokens in a text. It processes about 10,000 words/sec (here and below the cited processing speeds are very approximate, based on a 1Ghz Pentium running Linux with 2 Gbyte of RAM). The RASP parser (Briscoe and Carroll, 2002) is a statistically-trained parser which operates without a full lexicon (speed around 100 words/sec). The English Resource Grammar (ERG) (Copestake and Flickinger, 2000) can be processed by the LKB (Copestake, 2002) or PET (Callmeier, 2002). It incorporates a lexicon with detailed linguistic information. PET is highly optimised (5–30 words/sec, depending on the corpus) while the LKB is more suited for development and can be used for generation. PET, LKB and ERG are Open Source.

The ERG can produce more detailed RMRSs than RASP, but relies on a detailed lexicon to do this. For SciBorg, lexical information comes from the hand-built ERG lexicon, plus additional hand-constructed lexical entries for very common terms in Chemistry texts, plus unknown word handling based on OSCAR-3 (see below) and POS tags. Since the ERG cannot use the same tokeniser as OSCAR-3 or the RASP tagger, unknown word processing requires a rough match of text spans.

6 Domain-specific processing modules and resources

6.1 OSCAR-3

One area in which this architecture differs from more standard approaches is the role of the software which recognises entities such as chemical compounds in the text. Consider, for instance, the following text snippet:

```
We have recently communicated that
the condensation of
5-halo-1,2,3-thiadiazole-4-carboxylate(1)
with <it>o</it>-phenylenediamine(2)
affords thiadiazepine(3)
```

Domain-specific processing is required to deal with systematic chemistry names such as that of the first compound given here. Systematic names describe the structure of the compound, they are constructed productively, and new compounds are described very frequently, so this is not a matter of simply listing all names. The amaneunsis application needs to know what compound is being referred to: this is necessary to support search on particular compounds, for instance. In contrast, most domain-independent modules need to know that this stretch of text refers to some compound, but not the identity of the specific compound, since the linguistic

properties of a sentence are insensitive to chemical composition. Schematically, the section should appear to the domain-independent parsers as:

```
We have recently communicated that
the condensation of [compound-1]
with [compound-2] affords [compound-3]
```

However the output of the language processing component must be relatable to the specific identification of the compound for searches.

It is also important that the modules know about text regions which should not be subject to general morphological processing, lexical look-up etc. In particular, in data sections of papers, standard techniques for sentence splitting result in very large chunks of text being treated as a single sentence. Given that language processing has to be bounded by resource consumption, the expense of attempting to parse such regions could prevent analysis of 'normal' text. In our current project, the domain-specific processing is handled by OSCAR-3 (Corbett and Murray-Rust, 2006).

6.2 Ontologies and other domain-specific resources

In chemistry, the need for explict ontologies is reduced by the concept of molecular structure: structures and systematic names are (at least in principle) intercovertible, many classes of compounds (such as ketones) are defined by structural features, and structures may be related to each other using concepts such as isomerism and substructure-superstructure relationships that may be determined algorithmically. However, there are still many cases where explict ontology is required, such as in the mapping of trivial names to structures, and in the assignment of compounds to such categories as pesticides and natural products. The three ontologies for Chemistry that we are aware of are: Chemical Entities of Biological Interest (ChEBI: `www.ebi.ac.uk/chebi/index.jsp`); FIX (methods and properties: `obo.sourceforge.net/cgi-bin/detail.cgi?fix`) and REX (processes e.g., chemical reactions `obo.sourceforge.net/cgi-bin/detail.cgi?rex`). FIX and REX are currently development versions, while ChEBI has been fully released. ChEBI comes in two parts: a GO/OBO DAG-based ontology ('ChEBI ontology'), and a conventional structural database. This second half is used in OSCAR-3, providing a source of chemical names and structures.

The ChEBI ontology includes chemicals, classes of chemicals and parts of chemicals: these are organised according to structure, biological role and application. Unfortunately ChEBI does not explicitly distinguish between these types of entity. The application and biological role ontologies are currently relatively underdeveloped. However the ChEBI ontology has the potential to interact with other ontologies in the GO family.

The most useful other resource which is generally available is PubChem (`pubchem.ncbi.nlm.nih.gov`), which is designed to provide information on the biological activities of small molecules, but also provides information on their structure and naming. There is also the IUPAC Gold Book which is a compendium of chemical terminology with hyperlinks between entries. While not capable of directly supporting the IE needs mentioned in §2, these resources are potentially useful for lexicons and to support creation of ontology links.

7 Research markup and citation analysis

Searching in unfamiliar scientific literature is hard, even when a relevant paper has been identified as a starting point. One reason is that the status of a given paper with respect to similar papers is often not apparent from its abstract or the keywords it contains. For instance, a chemist might be more interested in papers containing direct experimental evidence rather than evidence by simulation, or might look for papers where some result is contradicted. Such subtle relationships between the core claims and evidence status of papers are currently not supported by search engines such as Google Scholar; if we were able to model them, this would add considerable value.

The best sources for this information are the papers themselves. Discourse analysis can help, via an analysis of the argumentation structure of the paper. For instance, authors typically follow the strategy of first pointing to gaps in the literature before describing the specific research goal – thereby adding important contrastive information in addition to the description of the research goal itself. An essential observation in this context is that conventional phrases are used to indicate the rhetorical status of different parts of a text. For instance, in Fig. 3 similar phrases are used to indicate the introduction of a goal, despite the fact that the papers come from different scientific domains.

Argumentative Zoning (AZ), a method introduced by Teufel (Teufel et al., 1999; Teufel and Moens, 2000), uses cues and other superficial markers to pick out important parts of scientific papers and supervised machine learning to find zones of different argumentative status in the paper. The zones are: AIM (the specific research goal of the current paper); TEXTUAL (statements about section structure); OWN (neutral descriptions of own work presented in current paper); BACKGROUND (gen-

Cardiology: The goal of this study was to elucidate the effect of LV hypertrophy and LV geometry on the presence of thallium perfusion defects. Heupler et al. (1997): 'Increased Left Ventricular Cavity Size, Not Wall Thickness, Potentiates Myocardial Ischemia', *Am Heart J*, 133(6)
Crop agriculture: The aim of this study was to investigate possible relationships between type and extent of quality losses in wheat with the infestation level of S. mosellana. Helenius et al. (1989): 'Quality losses in wheat caused by the orange wheat blossom midge Sitodiplosis mosellana', *Annals of Applied Biology*, 114: 409-417
Chemistry: The primary aims of the present study are (i) the synthesis of an amino acid derivative that can be incorporated into proteins /via/ standard solid-phase synthesis methods, and (ii) a test of the ability of the derivative to function as a photoswitch in a biological environment. Lougheed et al. (2004): 'Photomodulation of ionic current through hemithioindigo-modified gramicidin channels', *Org. Biomol. Chem*, Vol. 2, No. 19, 2798-2801
Natural language processing: In contrast, TextTiling has the goal of identifying major subtopic boundaries, attempting only a linear segmentation. Hearst (1997): 'TextTiling: Segmenting Text into Multi-paragraph Subtopic passages', *Computational Linguistics*, 23(1)
Natural language processing: The goal of the work reported here is to develop a method that can automatically refine the Hidden Markov Models to produce a more accurate language model. Kim et al. (1999): HMM Specialization with Selective Lexicalization, *EMNLP-99*

Figure 3: Similar phrases across the domains of chemistry and computational linguistics

erally accepted scientific background); CONTRAST (comparison with or contrast to other work); BASIS (statements of agreement with other work or continuation of other work); and OTHER (neutral descriptions of other researchers' work). AZ was originally developed for computational linguistics papers, but as a general method of analysis, AZ can and has been applied to different text types (e.g., legal texts (Grover et al., 2003) and biological texts (Mizuta and Collier, 2004)) and languages (e.g., Portuguese (Feltrim et al., 2005)); we are now adapting it to the special language of chemistry papers and to the specific search tasks in eChemistry. Progress towards construction of citation maps is reported in Teufel (2005) and Teufel et al. (2006).

The zones used in the original computational linguistics domain concentrated on the phenomena of attribution of authorship to claims (is a given sentence an original claim of the author, or a statement of a well-known fact) and of citation sentiment (does the author criticise a certain reference or use it as part of their own work). For application of AZ to chemistry, changes need to be made which mirror the different writing and argumentation styles in chemistry, in comparison to computational linguistics. Argumentation patterns are generally similar across the disciplines (they are there to convince the reader that the work undertaken is sound and grounded in evidence rather than directly carrying scientific information), but several factors such as the use of citations, passive voice, or cue phrases vary across domains.

For Chemistry, we intend to exploit the RMRS technology discussed earlier to detect cues. RMRS encoding is advantageous because it allows more concise and flexible specification of cues than do string-based patterns and because it allows identification of more complex cues. For instance, papers quite frequently explain a goal via a contrast using a phrase such as: *our goal is not to X but to Y*:

> our goal is not to verify P but to construct a test sequence from P[2]

Getting the contrast and scope of negation correct in such examples requires relatively deep processing. Processing of AZ cue phrases with ERG/PET should be feasible because their vocabulary and structure is relatively consistent.

8 Conclusion

The aim of this paper has been to describe how the separate strands of work on language processing within SciBorg fit together into a coherent architecture. There are many aspects of the project that we have not discussed in this paper because we have not yet begun serious investigation. This includes word sense disambiguation and anaphora resolution. The intention is to use existing algorithms, adapted as necessary to our architecture. We have also not discussed the application of Grid computing that will be necessary as we scale up to processing thousands of papers.

[2]Gargantini and Heitmeyer (1999), 'Using Model Checking to Generate Tests from Requirements Specifications' In Nierstrasz and Lemoine (eds), *Software Engineering - ESEC/FSE'99*, Springer

Acknowledgements

We are grateful to the Royal Society of Chemistry, Nature Publishing Group and the International Union of Crystallography for supplying papers. This work was funded by EPSRC (EP/C010035/1) with additional support from Boeing.

References

Briscoe, Ted, and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*.

Callmeier, Ulrich. 2002. Pre-processing and encoding techniques in PET. In Stephan Oepen, Daniel Flickinger, Jun'ichi Tsujii, and Hans Uszkoreit, eds., *Collaborative Language Engineering: a case study in efficient grammar-based processing*. Stanford: CSLI Publications.

Callmeier, Ulrich, Andreas Eisele, Ulrich Schäfer, and Melanie Siegel. 2004. The DeepThought Core Architecture Framework. In *Proc. of LREC-2004*.

Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications.

Copestake, Ann. 2003. Report on the design of RMRS. DeepThought project deliverable.

Copestake, Ann, and Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000)*, 591–600.

Copestake, Ann, Dan Flickinger, Ivan Sag, and Carl Pollard. 2005. Minimal Recursion Semantics: an introduction. *Journal of Research in Language and Computation* 3(2–3): 281–332.

Corbett, Peter, and Peter Murray-Rust. 2006. High-throughput identification of chemistry in life science texts. In *Proceedings of the 2nd International Symposium on Computational Life Science (CompLife '06)*. Cambridge, UK.

Feltrim, Valeria, Simone Teufel, G. Gracas Nunes, and S. Alusio. 2005. Argumentative Zoning applied to Critiquing Novices' Scientific Abstracts. In James G. Shanahan, Yan Qu, and Janyce Wiebe, eds., *Computing Attitude and Affect in Text*. Dordrecht, The Netherlands: Springer.

Frank, Anette, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, Brigitte Jörg, and Ulrich Schäfer. 2006. Question Answering from Structured Knowledge Sources. *Journal of Applied Logic, Special Issue on Questions and Answers: Theoretical and Applied Perspectives* 1.

Grover, Claire, Ben Hachey, and Chris Korycinsky. 2003. Summarising legal texts: Sentential tense and argumentative roles. In *Proceedings of the NAACL/HLT-03 Workshop on Automatic Summarization*.

Hearst, Marti A. 1992. Direction-Based Text Interpretation as an Information Access Refinement. In Paul S. Jacobs, ed., *Text-based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*. Hillsdale, NJ: Lawrence Erlbaum.

Hollingsworth, Bill, Ian Lewin, and Dan Tidhar. 2005. Retrieving Hierarchical Text Structure from Typeset Scientific Articles — a Prerequisite for E-Science Text Mining. In *Proc. of the 4th UK E-Science All Hands Meeting*, 267–273. Nottingham, UK.

Mizuta, Yoko, and Nigel Collier. 2004. An Annotation Scheme for Rhetorical Analysis of Biology Articles. In *Proceedings of LREC'2004*.

Ruland, Tobias, C. J. Rupp, Jörg Spilker, Hans Weber, and Karsten L. Worm. 1998. Making the Most of Multiplicity: A Multi-Parser Multi-Strategy Architecture for the Robust Processing of Spoken Language. In *Proc. of the 1998 International Conference on Spoken Language Processing (ICSLP 98)*, 1163–1166. Sydney, Australia.

Rupp, CJ, Ann Copestake, Simone Teufel, and Ben Waldron. 2006. Flexible Interfaces in the Application of Language Technology to an eScience Corpus. In *Proceedings of the 4th UK E-Science All Hands Meeting*. Nottingham, UK.

Rupp, C.J., J. Spilker, M. Klarner, and K.L. Worm. 2000. Combining Analyses from Various Parsers. In W. (ed.) Wahlster, ed., *Verbmobil: Foundations of Speech-to-Speech Translation*, 311–320. Berlin: Springer Verlag.

Teufel, Simone. 2005. Argumentative Zoning for improved citation indexing. In James G. Shanahan, Yan Qu, and Janyce Wiebe (Eds.), eds., *Computing Attitude and Affect in Text: Theory and Applications*, 159–170. Springer.

Teufel, Simone, Jean Carletta, and Marc Moens. 1999. An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of the Ninth Meeting of the European Chapter of the Association for Computational Linguistics (EACL-99)*, 110–117.

Teufel, Simone, and Marc Moens. 2000. What's yours and what's mine: Determining Intellectual Attribution in Scientific Text. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Teufel, Simone, Advaith Siddharthan, and Dan Tidhar. 2006. An annotation scheme for citation function. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*. Sydney, Australia.

Uszkoreit, Hans. 2002. New chances for deep linguistic processing. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*. Taipei, Taiwan.

Vlachos, Andreas, and Caroline Gasperin. 2006. Bootstrapping and Evaluating Named Entity Recognition in the Biomedical Domain. In *Proc. Proceeding of BioNLP (Poster session) in HLT-NAACL*. New York.

Waldron, Benjamin, and Ann Copestake. 2006. A Standoff Annotation Interface between DELPH-IN Components. In *Proceedings of the fifth workshop on NLP and XML (NLPXML-2006)*. Trento, Italy.

Waldron, Benjamin, Ann Copestake, Ulrich Schäfer, and Bernd Kiefer. 2006. Preprocessing and Tokenisation Standards in DELPH-IN Tools. In *Proceedings of LREC2006*. Genoa, Italy.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Flexible Interfaces in the Application of Language Technology to an eScience Corpus

C.J. Rupp, Ann Copestake, Simone Teufel, Benjamin Waldron

Computer Laboratory, University of Cambridge

Abstract

We describe two key interfaces used in an architecture for applying a range of Language Technology tools to a corpus of Chemistry research papers, in order to provide a basis of robust linguistic analyses for Information Extraction tasks. This architecture is employed in the context of the eScience project 'Extracting the Science from Scientific Publications' (a.k.a. SciBorg), as described in Copestake et al. (2006). The interfaces in question are the common representation for the papers, delivered in a range of formats, and the coding of various types of lingustic information as standoff annotation. While both of these interfaces are coded in XML their structure and usage are quite distinct. However, they are employed at the main convergence points in the system architecture. What they share is the ability to represent information from diverse origins in a uniform manner. We emphasise this degree of flexibility in our description of the interface structures and the design decisions that led to these definitions.

1 Introduction

The purpose of this paper is to document two interface structures that play a crucial role in the architecture of the eScience project 'Extracting the Science from Scientific Publications' (also known as SciBorg). The project's aims and the architecture itself are described in more detail in Copestake et al. (2006).

As the official project title suggests, SciBorg is concerned with Information Extraction (IE) from published scientific research, in this case Chemistry. However, the real challenge of the project lies in the comprehensive application of current Language Technology to an extensive corpus of research papers to provide robust linguistic analsyses, on which various Information Extraction tasks can be based. As we are, effectively, mining the text of Chemistry research, we feel it is appropriate to refer to the corpus of research papers as an *eScience corpus.*

In this context, the most significant interfaces are: the common representation for the papers as an input to the analysis tools and the pool of resulting analyses. We emphasise the flexibility required in both of these interfaces, but for different reasons. While the research papers may be delivered in a range of formats, an adequate common representation for both content and formatting information economises on the number of interfaces to be supported, as long as the cost of maintaining the converion processes can be kept to a minmum. In applying existing analysis tools, at different levels of analysis, we attempt to maximise the synergies between the available components, although the components themselves were not, necessarily, developed with this in mind. This means that interface structure that brings the analyses together must accommodate those interactions.

We adopt a common XML markup for the research papers and a standoff annotation formalism for the coding of various types of linguistic information. While both of these interfaces are coded in XML their structure and usage are quite distinct. What they share is the ability to represent information from diverse origins in a uniform manner.

2 The SciBorg Corpus

The SciBorg project is concerned with Information Extraction from published Chemistry research papers. The three publishers affiliated to the project: The Royal Society of Chemistry (RSC), Nature Publishing Group (NPG) and International Union of Crystallography (IUCr), have provided a corpus of recently published papers. Delivery is either in the form of XML, or (in the case of IUCr) in SGML that is easily converted to XML. However, the original XML encoding is specific to the publishers, following a DTD defined for their own needs. Obviously, there are far fewer interfaces to maintain, if we convert all the papers in the corpus to a common XML encoding, so that subsequent

processing modules only have to interface with one XML encoding. For this purpose we have adopted an XML schema that has been developed over the course of several projects for the precise purpose of representing the logical structure of scientific research papers. Nevertheless, some adaptations to the schema were require for the specific needs of a corpus collected in well-defined XML encodings of publishers' markup. We have named the result SciXML, intuitively XML for Science.

3 The Development of SciXML

SciXML originates in XML markup for the logical structure of scientific papers in Computational Linguistics (Teufel and Moens, 1997; Teufel, 1999). It has subsequently been employed to corpora from a variety of disciplines, including Cardiology (Teufel and Elhadad, 2002) and Genetics (Hollingsworth et al., 2005).

What is equally significant is that these corpora, while consistent in the function of their texts, were collected from a variety of different sources and in varying formats, so that conversions to SciXML have been defined from: LaTeX, HTML and PDF (via OCR). The conversion from low level formatting produced a cumulative effect on the immediate precursor for our SciXML schema. The more functional levels of the markup were impoverished, as only distinctions that affected the formatting could be retrieved. The handling of lists was rather simplified and tables excluded, because of the difficulty of processing local formatting conventions. Equally, the applications had no necessity to represent papers in full formatting, so information like the modification of font faces at the text level was excluded. While footnotes were preserved these were collected at the end of the paper, effectively as end notes, alongside the vestigial representations of tables and figures, chiefly their captions.

SciBorg has the advantage of access to publishers' markup which supports functional or semantic distinctions in structure and provides a detailed coding of the content and structure of lists and tables, rather than just their formatting on the page. However, we envisage IE applications which involve some rendering of the paper contents in a readable form, e.g. in authoring and proof reading aids for Chemists. While not as detailed as the publishers page formatting we would require the paper content to be recognisable, exploiting HTML as a cheap solution to any more complex display problems. This implies retaining more of the explicit formatting information, particularly at the text level. In practice, this has meant the addition of inline markup for font face selections and some inclusion of LaTeX objects for formulae, as well as the preservation of the origin points for floats, as they are known to LaTeX (table and figures). As a result SciXML retains a focus on the logical structure of the paper, but preserves as much formatting information as is required for effectively rendering the papers in an application.

4 The SciXML Schema

The resulting form of SciXML is defined as a Relax NG schema. Since this is an XML-based formalism, it is difficult to exhibit any substantive fragment in the space available here. Figure 1 shows just the overall strucuture of a paper and the first level of elements in the `<REFERENCELIST>` element. The most comprehensive description that is appropriate here is a catalogue of the types of construct in SciXML.

Paper Identifiers: We require enough information to identify papers both in processing and when tracing back references. While publisher's markup may contain an extensive log of publication and reviewing, a handful of elements suffice for our needs: `<TITLE>`, `<AUTHOR>`, `<AUTHORS>`, `<FILENO>`, `<APPEARED>`

Sections: The hierarchical embedding of text segments is encoded by the `<DIV>` element, which is recursive. A `DEPTH` attribute indicates the depth of embedding of a division. Each division starts with a `<HEADER>` element.

Paragraphs are marked as element `<P>`. The paragraph structure has no embedding in SciXML, but paragraph breaks within `<ABSTRACT>`, `<CAPTION>` and `<FOOTNOTE>` elements and within list items (`<LI>`) are preserved noted with a `<SUBPAR>` element.

Abstract, Examples and Equations are text sections with specific functions and formatting and can be distinguished in both publishers' markup and in the process of recovering information from PDF. We have added a functionally determined section `<THEOREM>`, as we have encountered this type of construct in some of the more formal papers. Similarly, linguistic examples were distinguished early on, as Computational Linguistics was the first corpus to be treated.

Tables, figures and footnotes are collected in a list element at the end of the document, `<TABLELIST>`, `<FIGURELIST>`, `<FOOTNOTELIST>`, respectively. The textual position of floats is marked by an `<XREF/>` element. The reference point of a footnote is marked by a separate series of markers: `<SUP>`, also used for similar functions in associating authors and affiliations.

```
<define name="PAPER.ELEMENT">
  <element name="PAPER">
    <ref name="METADATA.ELEMENT" />
    <optional>
      <ref name="PAGE.ELEMENT" />
    </optional>
    <ref name="TITLE.ELEMENT" />
    <optional>
      <ref name="AUTHORLIST.ELEMENT" />
    </optional>
    <optional>
      <ref name="ABSTRACT.ELEMENT" />
    </optional>
    <element name="BODY">
      <zeroOrMore>
        <ref name="DIV.ELEMENT" />
      </zeroOrMore>
    </element>
    <optional>
      <element name="ACKNOWLEDGMENTS">
        <zeroOrMore>
          <choice>
            <ref name="REF.ELEMENT" />
            <ref name="INLINE.ELEMENT" />
          </choice>
        </zeroOrMore>
      </element>
    </optional>
    <optional>
      <ref name="REFERENCELIST.ELEMENT">
    </optional>
    <optional>
      <ref name="AUTHORNOTELIST.ELEMENT">
    </optional>
    <optional>
      <ref name="FOOTNOTELIST.ELEMENT">
    </optional>
    <optional>
      <ref name="FIGURELIST.ELEMENT">
    </optional>
    <optional>
      <ref name="TABLELIST.ELEMENT">
    </optional>
  </element>
</define>

<define name="REFERENCELIST.ELEMENT">
  <element name="REFERENCELIST">
    <zeroOrMore>
      <ref name="REFERENCE.ELEMENT" />
    </zeroOrMore>
  </element>
</define>
```

Figure 1: A fragment of the Relax NG schema for SciXML

Lists: various types of lists are supported with bullet points or enumeration, according to the TYPE attribute of the <LIST> element. Its contents will be uniformly marked up as <LI> for list items.

Cross referencing takes a number of forms including the <SUP> and <XREF> mentioned above. All research papers make use textual cross references to identify sections and figures. For Chemistry, reference to specific compounds was adopted, from the publishers' markup conventions. The other crucial form of cross reference in research text is citations, linking to bibliographic information in the <REFERENCELIST>.

Bibliography list: The bibliography list at the end is marked as <REFERENCELIST>. It consists of <REFERENCE> items, each referring to a formal citation. Within these reference items, names of authors are marked as <SURNAME> elements, and years as <YEAR>.

5 The Conversion Process

The conversion from the publisher's XML markup to SciXML can be carried out using an XSLT stylesheet. In fact, most of the templates are quite simple, mainly reorganising and/or renaming content. The few exceptions are collections of elements such as footnotes and floats to list elements and replacing the occurrence *in situ* with a reference marker. Elements that explicitly encode the embedding of text divisions are systematically mapped to a non-hierarchical division with a DEPTH attribute recording the level of embedding. Figure 2 shows a template for converting a section (<sec>) element

```
<xsl:template match="sec">
<DIV DEPTH="{@level}">
<xsl:apply-templates/>
</DIV>
</xsl:template>
```

Figure 2: An XSLT conversion template

from a publisher's markup to a <DIV> in SciXML. The SciXML conventions show an affinity for the structure of the formatted text which follows directly from usage with text recovered from PDF. For our current purposes this is considered harmless, as no information is lost. We assume that an application will render the content of a paper, as required, e.g. in HTML. In fact, we already have demonstration applications that systematically render SciXML in HTML via an additional stylesheet. The one SciXML convention that is somewhat problematic is the flattening of paragraph structure. While divisions can embed divisions, paragraphs may not embed paragraphs, not even indirectly. To avoid information loss, here an empty paragraph break marker is added to list elements, abstracts and footnotes which may, in the general case, include paragraph divisions. Although earlier versions of SciXML also encoded sentence boundaries, the sentence level of annotation has been transferred to the domain of linguistic standoff annotation, where it can be generated by automatic sentence splitters.

While the XSLT stylesheets that convert the publisher XML to SciXML are relatively straightfor-

ward, each publisher DTD or schema requires a separate script. This places a manual overhead on the inclusion of papers from a new publisher, but fortunately at a per publisher rather than per journal level. In practice, we have found that the element definitions are still sufficiently similar to allow a fair amount of cut-and-paste programming, so that the overhead decreases as more existing conversion templates exist to draw on. A recent extension of SciXML conversion to the PLoS DTD presented remarkably few unknown templates.

6 Language Technology in SciBorg

The goals of the SciBorg project, as its full title suggests, concern Information Extraction, in fact a range of IE applications are planned all starting from a corpus of published chemistry research. However, the common path to those goals and the real challenge of the project is the application of Language Technology, and, in particular, linguistically motivated analysis techniques. In practice, this involves the use of so-called 'deep' parser, based on detailed formal descriptions of the language with extensive grammars and lexicons, and shallower alternatives, typically employing stochastic models and the results of machine learning. This multi-engine approach has been employed in Verbmobil (Ruland et al., 1998; Rupp et al., 2000) and Deep Thought (Callmeier et al., 2004).

While there have been considerable advances in the efficiency of deep parsers in recent years, there are a variety of ways that performance can be enhanced by making use of shallower results, e.g. as preprocessors or as a means of selecting the most interesting sections of a text for deeper analysis. In fact, the variety of different paths through the analysis architecture is a major constraint on the design of our formalism for linguistic annotations, and the one which precludes the use of the major existing frameworks for employing Language Technology in IE, such as GATE (`http://gate.ac.uk`).

7 Multiple Analysis Components

The main deep and shallow parsing components that we use have been developed over a period of time and represent both the state of the art and the result of considerable collaboration.

PET/ERG: PET (Callmeier, 2002) is a highly optimise HPSG parser that makes use of the English Resource Grammar (ERG) (Copestake and Flickinger, 2000). The ERG provides a detailed grammar and lexicon for a range of text types. The coverage of the PET/ERG analysis engine can be extended by an unknown word mechanism, provided that a partial identification of the word class is possible, e.g. by POS (part of speech) tagging.

RASP provides a statistically trained parser that does not require a full lexicon (Briscoe and Carroll, 2002). The parser forms part of a sequence of analysers including a sentence splitter, tokeniser and a POS tagger.

A key factor in combining results from multiple parsers is that they present compatible results. Both of the parsers we are using are capable of producing analyses and partial analyses in the same form of representation: Robust Minimal Recursion Semantics (Copestake, 2003), henceforth RMRS. This is a form of underspecified semantics resulting from the tradition of underspecification in symbolic Machine Translation. In RMRS, the degree of underspecification is extended so that all stages of deep and shallow parsing can be represented in a uniform manner. It is therefore feasible to combine the results of the deep and shallow parsing processes. Our presentation of the parsers above should have suggested one other path through the system architecture, in that the RASP tagger can provide the information necessary to run the unknown word mechanism in the PET parser, what this requires is a mapping from the part of speech tag to an abstract lexical type in the ERG grammar definition.

The combination of results from deep and shallow parsers is only part of the story, as these are general purpose parsers. We will also need components specialised for research text and, in particular, for Chemistry. The specialised nature of the text of Chemistry research is immediately obvious, both in specialised terms and in sections which are no longer linguistic text, as we know it. A sophisticated set of tools for the analysis of Chemistry research is being developed within the SciBorg project, on the basis of those described in Townsend et al. (2005). These range from NER (Named Entity Recognition) for Chemical terms, through the recognition of data sections, to specialised Chemistry markup with links to external and automatically generated information sources. For the general parsing task the recognition of specialised terms and markup of data sections are the most immediate contribution. Though this does, to some extent, exacerbate the problems of ambiguity, as now deep, shallow and Chemistry NER processes have different results for the same text segment.

The overlap between ordinary English and Chemical terms can be trivially demonstrated, in as much as the prepositions *in* and *as* in sentence initial position can be easily confused with *In* and *As*, the chemical symbols for indium and arsenic, respec-

tively. Fortunately some English words that moonlight as chemical symbols are less common in research text, such as *I*, but this is only an example of the kinds of additional ambiguity. Formally, the word *lead* would have at least 3 analyses, as verb, noun and element, but the latter two are not orthogonal.

8 Standoff Annotation

Standoff annotation is an increasingly common solution for pooling different types of information about a text. Essentially, this entails a separation of the original text and the annotations into two separate files. There are some practical advantages in this: you are less likely to obscure the original text under a mesh of annotations and less likely to proliferate numerous partially annotated versions of the object text. However, the true motivation for standoff annotation is whether different annotation schemes will impose different structures on the text. To some extent XML forces a standoff annotation scheme, by enforcing a strict tree structure, so that even one linguistic annotation of a formatted text above the word level risks becoming incompatible with the XML DOM tree. As a simple example, we exhibit here a fragment of formatted text from a data section of a Chemistry paper, alongside its SciXML markup and a simple linguistic markup for phrase boundaries.

Formatted text *calculated for* C_{11} H_{18} O_3

SciXML markup <**it**> calculated for </**it**>
C<sb>11</sb>H<sb>18</sb>O<sb>3</sb>

Phrasal markup <v>calculated</v>
<**pp**>for <ne>C11H1803</ne></**pp**>

Here, the assignment of a simple phrase structure conflicts with the formatting directives for font face selection. The XML elements marked in bold face cannot be combined in the same XML dominance tree.

With each additional type or level of annotation the risk of such clashes increases, so a clean separation between the text and its annotations is helpful, but where do you separate the markup and how do you maintain the link between the annotations and the text they address. As we have a common XML markup schema, including logical structure and some formatting information we have a clear choice as to our text basis. The link between text and annotations is usually maintained by indexing. Here, there are some options available.

8.1 Indexing

The indexing of annotations means that each element of the standoff file encodes references to the position in the text file where the text it relates to occurs, typically this encodes a span of the text. Some form of indexing is a prerequisite for standoff annotation. The simplest indexing is by byte offset, encoding the position of a text segment in terms of the number of bytes in the file preceding its start and finish positions. This is universal, in that it can cope with all types of file content, but is not stable under variations in character encoding. For an XML source text, indexing by character offset is more useful, particularly if character entities have been resolved. Extraneous binary formats such as graphics will be encoded externally as <NDATA> elements. In fact, the variant of character offset we adopt involves numbering the Unicode character points, as shown in Figure 3.

Raw Text: " Come here! "

Unicode character points:

```
.<.p.>.C.o.m.e. Δ .<. i .>. h .e.r.e .
0 1 2 3 4 5 6 7   8   9  10 11  12 13 14  15
                <. / . i .>. ! .<./.p.>.
                  16 17 18 19 20 21 22 23  24
```

Figure 3: character pointers (points shown as '.')

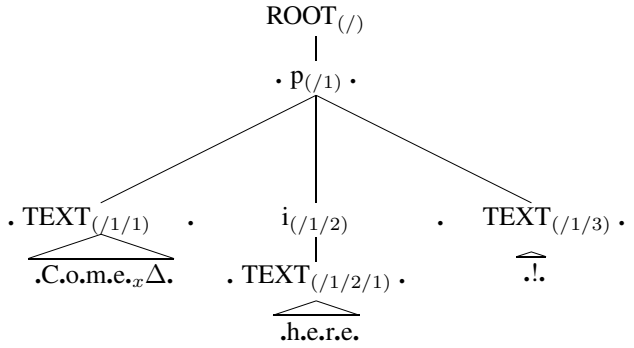
A simple annotation of linguistic tokens would then produce three elements:

```
<w from='3' to='7'>Come</w>
<w from='11' to='15'>here</w>
<w from='19' to='20'>!</w>
```

XML also offers an additional structure for navigating around the file contents.We can take the XML DOM trees as a primary way of locating points in the text and character positions as secondary, so that a character position is relative to a branch in the XML tree. This XPoint notation is demonstrated in Figure 4. The XML tree positions will not be affected by changes within an XML element, e.g. additional attributes, but will not be stable against changes in the DOM structure itself.

We currently make use both character offset indexing and XPoint indexing in different modules. This requires conversion via a table which relates XPoints to character positions for each file. The initial choice of indexing mode for each subprocess was influenced by the availability of different styles of XML parser[1], but also informed by the ease of converting character information from non-XML linguistic tools. In the long term our aim would be to eliminate the use character offset indexing.

[1] Computing character offsets is easier with an event-based parser that has access to the input parsed for each event. XPoints can be tracked more easily in a DOM parser.



xpoint at x is: "`/1/1.4`"

Figure 4: xpoint-based pointers (points shown as '.')

9 Types of Annotation

We have motivated standoff annotation on the basis of the incompatibility between the tree structure of an XML document and the annotation of arbitrary segments of that document. The collection of annotations can be encoded in an XML format with its own DTD, because they form, together, the representation of a grap structure, as a set of edges. The nodes of this graph are, ultimately, the character positions in the original XML document, whichever way you index them. This is similar to a chart or well-formed substring table in a parser, except that muliple tokenisations can lead to varying word boundaries. We therefore have more potential nodes in the annotation graph than a (single) parser's chart. In contrast, a word lattice, as used in speech recognition, may have still more potential nodes. In a word lattice, the succession of nodes is not only determined by a physical convergence, but also by factors like statistical language models. These additional constraints mean that a word lattice may have distinct nodes that represent the same time frame. While this condition will not occur in our annotation graphs, we have adopted a standard for the DTD of our graphs which is general enough to support the treatment of word lattice inputs in speech processing. The reason for this is to link up with existing standards in the Language Technology community, and, in particular, in the HPSG processing comunity.

The origin of our annotation standard lies with the (ISO working draft) MAF standard (Clement and de la Clergerie, 2005) for morphological annotations. A variant of this had been developed within the DELPH-IN community as an emergent standard for input to parsers. This is known as SMAF (Waldron et al., 2006). Our annotation graphs collect all the annotations for a whole document in one file. This means that we require require a further generalisation on the SMAF standard. We have termed this SAF (Waldron and Copestake, 2006). We also encode analysis results, as well as input sentences, tokens and tagging, in the same annotation file. Although these various annotations have the same general form and the content of an annotation is essentially based on RMRS, the details for each type of annotation vary slightly.

9.1 Sentences

The results of a sentence splitter are represented by an annotation edge with the `type` sentence, unique identifier, initial and final index positions, initial and final lattice nodes and the text content of the sentence, as a `value` attribute string.

```
<annot type='sentence' id='s133'
from='42988' to='43065'
source='v4987' target='v5154'
value='calculated for C11H18O3'/>
```

9.2 Tokens

A tokeniser determines tokens at the word level, including punctuation and common abbreviations. These are represented by annotations of the `type` token, with a span in terms of offset positions and lattice nodes, as well as a dependency and the token string as a `value`.

```
<annot type='token' id='t5153'
from='43035' to='43065'
source='v5152' target='v5153'
deps='s133' value='C11H18O3'/>
```

We record a dependency between the tokenisation and sentence splitting because of the sequence of processing in the SciBorg architecture.

9.3 POS Tags

Part of speech tags are annotated as being dependent on a tokenisation, this is a fixed relation between the two processing steps and it allows us some economy in representing the annotation.

```
<annot type='pos' id='p5153' deps='t5153'
source='v5152' target='v5153' value='NP1'/>
```

```
<annot type='rmrs' id='r2' from='42988' to='43065' source='v5150' target='v5153'>
<rmrs cfrom='42988' cto='43043'>
<label vid='1'/>
<ep cfrom='42988' cto='43030'>
  <realpred lemma='calculate' pos='v' sense='1'/><label vid='10'/><var sort='e' vid='2'/></ep>
<ep cfrom='43031' cto='43034'>
  <realpred lemma='for' pos='p'/><label vid='10001'/><var sort='e' vid='13'/></ep>
<ep cfrom='43035' cto='43065'>
  <gpred>proper_q_rel</gpred><label vid='14'/><var sort='x' vid='12'/></ep>
<ep cfrom='-1' cto='-1'>
  <gpred>named_rel</gpred><label vid='17'/><var sort='x' vid='12'/></ep>
<rarg><rargname>ARG2</rargname><label vid='10'/><var sort='x' vid='3'/></rarg>
<rarg><rargname>ARG1</rargname><label vid='10001'/><var sort='e' vid='2'/></rarg>
<rarg><rargname>ARG2</rargname><label vid='10001'/><var sort='x' vid='12'/></rarg>
<rarg><rargname>RSTR</rargname><label vid='14'/><var sort='h' vid='15'/></rarg>
<rarg><rargname>BODY</rargname><label vid='14'/><var sort='h' vid='16'/></rarg>
<rarg><rargname>CARG</rargname><label vid='17'/><constant>*TOP*</constant></rarg>
<ing><ing-a><var sort='h' vid='10'/></ing-a><ing-b><var sort='h' vid='10001'/></ing-b></ing>
<hcons hreln='qeq'><hi><var sort='h' vid='15'/></hi><lo><label vid='17'/></lo></hcons>
</rmrs>
</annot>
```

Figure 5: An RMRS annotation

Using the RASP tagger, the tagset is based on CLAWS. The POS tagger also provides the option of an RMRS output.

9.4 Chemical Terms

We use the NER functionality in OSCAR-3 (Corbett and Murray-Rust, 2006) to provide an annotation for Chemical terms.

```
<annot type="oscar" id="o554"
from="/1/5/6/27/51/2/83.1"
to="/1/5/6/27/51/2/88/1.1" >
  <slot name="type">compound</slot>
  <slot name="surface">C11H18O3</slot>
  <slot name="provenance">formulaRegex</slot>
</annot>
```

This example differs from the annotations shown above in two obvious respects: the indexing is given in XPoint rather than character offsets and the XML element has content rather than a `value` attribute. This representation provides information about the way that the named entity recognition was arrived at. For use in further stages of analysis, this content should be made compatible with an RMRS representation.

9.5 RMRS Annotations

For the sake of completeness we include an example of an RMRS annotation in Figure 5. This represents the result for a partial analysis. The content of this element is represented in the XML form of RMRS annotation. However, the mechanisms which make RMRS highly suitable for representing arbitrary levels of semantic underspecification in a systematic and extensible way, e.g. allowing monotonic extension to a more fully specified RMRS, up to the representation of a full logical formula, make this representation relatively complex. Copestake (2003) provides a detailed account of the RMRS formalism.

10 SAF as a Common Interface

A SAF file pools linguistic annotations from all levels of language processing and from distinct parsing strategies. In this role it provides a flexible interface in a heterarchical processing architecture that is not committed to a single pipelined path through a fixed succession of processes. While this is reminiscent of a blackboard architecture or more localised pool architectures, it is only a static representation of the linguistic annotations. It does not provide any specific mechanisms for accessing the information in the annotations, nor does it require any particular communications architecture in the processing modules. In fact, there may have been more efficient ways of representing a graph or lattice of linguistic annotations, if it were not for the fact that the linguistic components we employ were defined in a range of different programming languages, so that an external XML interface structure has priority over any shared data structure. Given this fact, SAF is a highly appropriate choice for the common interface that is a key feature of any multi-engine analysis architecture.

11 Conclusions

We have presented two interface structures used in the architecture of the SciBorg project. Each of these occupies a crucial position in the architecture. SciXML provides a uniform XML encoding for research papers from various publishers, so that all subsequent language processing only has to interface with SciXML constructs. SAF provides a common representation format for the results of various Language Technology components. We have emphasised the flexibility of these interfaces. For SciXML, this consists in conversion from publishers' markup following a range of DTDs, as well as

other formats, including HTML, LaTeX and even PDF. For SAF, the primary mark of flexibility is in allowing partial results from multiple parsers to be combined at a number of different levels.

12 Acknowledgements

We are very grateful to the Royal Society of Chemistry, Nature Publishing Group and the International Union of Crystallography for supplying papers. This work was funded by EPSRC (EP/C010035/1) with additional support from Boeing.

References

Briscoe, Ted, and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*.

Callmeier, U., A. Eisele, U. Schäfer, and M. Siegel. 2004. The DeepThought Core Architecture Framework. In *Proceedings of LREC-2004*, 1205–1208. Lisbon, Portugal.

Callmeier, Ulrich. 2002. Pre-processing and encoding techniques in PET. In Stephan Oepen, Daniel Flickinger, Jun'ichi Tsujii, and Hans Usz koreit, eds., *Collaborative Language Engineering: a case study in efficient gra mmar-based processing*. Stanford: CSLI Publications.

Clement, L., and E.V. de la Clergerie. 2005. MAF: a morphosyntactic annotation framework. In *Proceedings of the 2nd Language and Technology Conference*. Poznan, Poland.

Copestake, Ann. 2003. Report on the design of RMRS. DeepThought project deliverable.

Copestake, Ann, Peter Corbett, Peter Murray-Rust, C. J. Rupp, Advaith Siddharthan, Simone Teufel, and Ben Waldron. 2006. An Architecture for Language Technology for Processing Scientific Texts. In *Proceedings of the 4th UK E-Science All Hands Meeting*. Nottingham, UK.

Copestake, Ann, and Dan Flickinger. 2000. An open-source grammar development environment and broad-cover age English grammar using HPSG. In *Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000)*, 591–600.

Corbett, Peter, and Peter Murray-Rust. 2006. High-throughput identification of chemistry in life science texts. In *Proceedings of the 2nd International Symposium on Computational Life Science (CompLife '06)*. Cambridge, UK.

Hollingsworth, Bill, Ian Lewin, and Dan Tidhar. 2005. Retrieving Hierarchical Text. 2005. Structure from Typeset Scientific Articles - a Prerequisite for E-Science Text Mining. In *In Proceedings of the 4th UK E-Science All Hands Meeting*, 267–273. Nottingham, UK.

Ruland, Tobias, C. J. Rupp, Jörg Spilker, Hans Weber, and Karsten L. Worm. 1998. Making the Most of Multiplicity: A Multi-Parser Multi-Strategy Architecture for the Robust Processing of Spoken Language. In *Proc. of the 1998 International Conference on Spoken Language Processing (ICSLP 98)*, 1163–1166. Sydney, Australia.

Rupp, C. J., Jörg Spilker, Martin Klarner, and Karsten Worm. 2000. Combining Analyses from Various Parsers. In Wolfgang Wahlster, ed., *Verbmobil: Foundations of Speech-to-Speech Translation*, 311–320. Berlin: Springer-Verlag.

Teufel, S., and N. Elhadad. 2002. Collection and linguistic processing of a large-scale corpus of medical articles. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*.

Teufel, Simone. 1999. Argumentative Zoning: Information Extraction from Scientific Text. Ph.D. thesis, School of Cognitive Science, University of Edinburgh, Edinburgh, UK.

Teufel, Simone, and Marc Moens. 1997. Sentence extraction as a classification task. In Inderjeet Mani and Mark T. Maybury, eds., *Proceedings of the ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization*, 58–65.

Townsend, Joe, Ann Copestake, Peter Murray-Rust, Simone Teufel, and Chris Waudby. 2005. Language Technology for Processing Chemistry Publications. In *Proceedings of the fourth UK e-Science All Hands Meeting (AHM-2005)*. Nottingham, UK.

Waldron, Benjamin, and Ann Copestake. 2006. A Standoff Annotation Interface between DELPH-IN Components. In *The fifth workshop on NLP and XML: Multi-dimensional Markup in Natural Language Processing (NLPXML-2006)*.

Waldron, Benjamin, Ann Copestake, Ulrich Schäfer, and Bernd Kiefer. 2006. Preprocessing and Tokenisation Standards in DELPH-IN Tools. In *Proceedings of the 5th International Conference on Language Re sources and Evaluation (LREC-2006)*. Genoa, Italy.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

The CLEF Chronicle: Transforming Patient Records into an E-Science Resource

Jeremy Rogers, Colin Puleston, **Alan Rector**
Biohealth Informatics Group, University of Manchester, UK
jeremy.e.rogers@manchester.ac.uk
puleston@cs.man.ac.uk
rector@cs.man.ac.uk

Abstract

Electronic patient records are typically optimised for delivering care to a single patient. They omit significant information that the care team can infer whilst including much of only transient value. They are secondarily an indelible legal record, comprised of heterogeneous documents reflecting local institutional processes as much as, or more than, the course of the patient's illness.

By contrast, the CLEF Chronicle is a unified, formal and parsimonious representation of how a patient's illness and treatments unfold through time. Its primary goal is efficient querying of aggregated patient data for clinical research, but it also supports summarisation of individual patients and resolution of co-references amongst clinical documents. It is implemented as a semantic network compliant with a generic temporal object model whose specifics are derived from external sources of clinical knowledge organised around ontologies.

We describe the reconstruction of patient chronicles from clinical records and the subsequent definition and execution of sophisticated clinical queries across populations of chronicles. We also outline how clinical simulations are used to test and refine the chronicle representation. Finally, we discuss a range of engineering and theoretical challenges raised by our work.

1. Introduction

The Clinical e-Science Framework (CLEF) project is a United Kingdom eScience project, sponsored by the Medical Research Council. It aims to establish a policy and technical infrastructure through which data arising from routine medical care across multiple sites and institutions may be collected and presented as an aggregated research repository in support of biomedical research [1].

Existing patient records are very rich. A typical example in our research corpus contains a thousand or more numeric data points, a chronology of five or six hundred significant clinical events (clinics attended, drugs dispensed, etc.), plus a couple of hundred narrative documents (letters between doctors, pathology, radiology or body scan results etc.). However, even when in electronic form, most of this data is intended for exclusively human interpretation. Furthermore, much of the critical information is left implicit. To take a common example from the cancer domain, it is rarely stated explicitly why the drug Tamoxifen is being given. Clinicians know that there is little other reason to prescribe it other than to prevent recurrence of breast cancer and, as a general rule, do not bother to record what can be assumed. Similarly, when a drug must be stopped because of its side effects (e.g. anaemia due to chemotherapy), the causal connection between side effect and stopping the drug is rarely mentioned, and even the nature of the side effect itself may have to be inferred from e.g. serial laboratory tests showing low haemoglobin values rather than being explicitly stated as 'anaemia'.

Other partners in the CLEF programme are researching technologies to retrieve useful information from clinical narratives [2]. This paper is a progress report on an orthogonal problem: assuming the full richness of clinical information were available – whether extracted from traditional clinical records *post hoc* or acquired *a priori* using entirely different clinical data capture paradigms – how might that information be represented for the maximal benefit of clinical research? Clinicians are particularly interested in

certain types of information rarely formalised in current records, *e.g.*

- the precise nature of any imaging investigation (e.g. which anatomical structure was imaged)
- the result of any investigation (what was found, what it signifies)
- why any investigation was ordered
- earlier provisional diagnoses
- what symptoms the patient experienced from the disease, or its treatment
- when treatment was changed and why
- abstractions such as 'relapse' or 'stage' of disease

Our goal is to define a representation supporting increasingly detailed information about patients and allowing us to evaluate its usefulness. We call this novel representation the CLEF Chronicle.

2. Properties of CLEF Chronicle

The CLEF Chronicle for an individual patient seeks to represent their clinical story entirely as a network of typed instances and their interrelations. Figure 1 illustrates the general flavour of what we are trying to represent: a patient detects a painful mass in their breast, as a result of which a clinic appointment occurs, where drug treatment for the pain is arranged and also a biopsy of the (same) mass in the (same) breast. The first clinic arranges a follow-up appointment, to review the (same) biopsy, which finds cancer in the (same) mass. This (same) finding is in turn the indication for radiotherapy to the (same) breast.

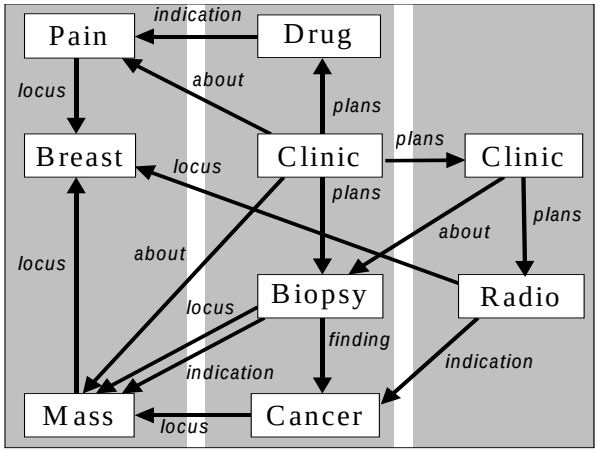


Figure 1: Informal View of Patient-History Fragment
(NOTE: Time-flow is *roughly* left → right)

In addition to the obvious structural difference between this representation and that of traditional electronic records, any clinical content represented as a CLEF Chronicle should have two central properties:

Parsimony – traditional patient records contain multiple discrete mentions of relevant instances in the real world (the tumour, the breast etc). A CLEF Chronicle should have only one occurrence of each.

Explicitness – traditional patient records may only imply clinically important information (e.g. the fact of relapse). This must be explicit within a CLEF Chronicle Representation.

3. Functions of CLEF Chronicle

Firstly, and most importantly, the CLEF Chronicle is intended to support more detailed, and more expressive, querying of aggregations of patient stories than is currently possible whilst at the same time improving the efficiency of complex queries. More detailed, because the Chronicle is more explicit: we can now ask e.g. how many patients 'relapsed' within a set period of treatment. More expressive, because the typing information associated with each Chronicle instance is drawn from a rich clinical ontology, such that queries may be framed in terms of arbitrarily abstract concepts: we can ask how many cancers of the lower limb were recorded, and expect to retrieve those of all parts of the lower limb. This is more efficient, because the traditional organization of patient records tends to require much serial or nested processing of records.

Secondly, an individual Chronicle can serve as an important knowledge resource during its own reconstruction from available electronic sources of traditional clinical records. In particular, a Chronicle can help resolve the frequent co-references and repeated references to real-world instances such as characterize traditional records. For example, heuristic and other knowledge linked to an ontology of Chronicle data types can be used to reject any request to instantiate more than one identifier for a [Brain], or any attempt to merge two mentions of [Pregnancy] separated by more than 10 months.

Thirdly, the Chronicle is intended to serve as a knowledge resource from which summarising information or abstractions may be inferred. For example, deducing 'anaemia' from a run of discrete low blood counts obtained from the traditional record, or 'remission' from several years of clinical inactivity and 'relapse' when this is followed by a flurry of tests and a new course of chemotherapy. Similarly, where the record does not explicitly say why drug X was given, reasoners browsing a Chronicle may identify

condition Y because the drug has no other plausible context of use.

Fourthly, the CLEF Chronicle is intended to support automatic summarisation of patient records. Given the sometimes chaotic nature of real patient records, manual case summarisation is recognised as good clinical practice. Manual derivation of such summaries from the content of the record is, however, notoriously time consuming whilst the result of such labours is notoriously out of date whenever it would be most clinically valuable.

The fundamental problem is that existing patient records are really "logbooks" (with similar legal significance) of what healthcare staff have heard, seen, thought and done [3] They often contain contradictory information; each entry reflects the understanding of the problem at the time it was made so that tracing the evolution of problems is non-trivial. The information is recorded in the order in which it was discovered rather than in the order in which it occurred – hence later entries may often include information about earlier events, and a precise diagnosis may not be available until long after the first entries that pertain to it were made. Much of the information is either un-interpreted (for example the serial low haemoglobins already described) or under-interpreted: by convention, radiologists only report what they are confident they have seen on an image. Thus, a typical xray report may state only 'There is evidence of moderate osteoporosis in the bony spine' thus leaving it to the requesting physician to infer the more important additional interpretation '(but) there are no osteolytic lesions that might suggest to cancer has spread to the bone'.

Maintaining summaries of clinical records, therefore, requires them to be transformed from a log whose chronology reflects the time of discovery of the underlying data to one reflecting the order of occurrence of events. Further, the chronology must be recorded at an appropriate level of temporal abstraction such that we may infer our best guest at what happened and how and why the patient was managed.

A CLEF Chronicle is intended to be a valuable substrate from which custom summaries of the story may be generated as human readable text. Different summarization strategies (and display vocabularies) can be used for different classes of user, such as for the doctor or the patient.

4. Chronicle Representation

The *Chronicle Representation* comprises a collection of core concepts represented as a Java-based *Chronicle Object Model (COM),* and more detailed knowledge represented via a set of declarative *External Knowledge Sources (EKS)*. This dual scheme is determined by the differing characteristics of the represented concepts. The architecture also allows for the incorporation of *EKS-related inference mechanisms* that can assist in the dynamic expansion of the *COM* (see below).

The concepts represented in the *COM* require associated procedural code of a concept-specific nature. The object-oriented format provides a natural means of achieving this. Though the representation of such concepts might alternatively be divided into declarative and procedural sections, this would result in an inelegant duplication of the model structure. Moreover, we believe the *COM* will be relatively stable and generic, such that ease-of-update and flexibility are not primary considerations.

Conversely, the *EKS* contain detailed knowledge requiring more regular maintenance, but are less likely to require concept-specific processing. Hence, a declarative format is more appropriate.

The precise division between *COM* and *EKS* is pragmatic rather than principled. Details are likely to evolve as the *Chronicle Representation* develops, with concepts crossing the line in both directions.

We envisage the *EKS* as a collection of knowledge sources, ultimately provided by multiple different ontologies, databases and sets of medical archetypes [4]. Similarly, the *EKS-related inference mechanisms* could be of varying types, including logic-based reasoning systems, rule-base systems, and dedicated procedural mechanisms. The *COM* is indifferent to how components of the *EKS* are represented, and how associated inferences are achieved, with both representations and inference mechanisms being accessed via a suitable Java API. Currently, the *EKS* for the patient chronicle is provided by a single OWL ontology developed specifically to meet our immediate requirements, and the associated inference by a Description Logic [6] reasoning system (FaCT++ [9]).

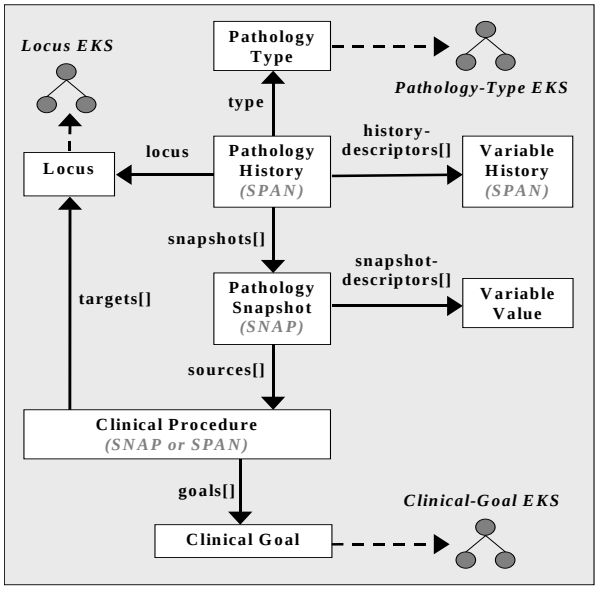


Figure 2: Chronicle Object Model (*COM*) Fragment (Simplified Version)

The *COM* has been developed as a standard Java-style API, primarily for use by the 'chroniclisation' mechanisms that we will develop, and the 'chronicle simulator' (described below). However, a more generic network-style representation, together with an associated translation mechanism based on the Java introspection facility, is available. This alternative representation provides a means of representing *COM*-based queries, and is suitable for driving chronicle-display and query-formulation GUIs.

The *COM* comprises both a generic section and a specific clinical section, with the latter being built upon the former. The generic section includes, most notably, the following:

- *EKS* access facilities;
- A temporal model that implements Smith's SNAP/SPAN distinction [5].

Figure 2 depicts a fragment of the clinical section. It can be seen that the representation of a particular pathology (such as an individual tumour or headache) comprises two main elements:

- A single *PathologyHistory* object, representing the pathology as a SPAN entity through time.
- An associated set of *PathologySnapshot* objects, each representing a SNAP view of the pathology at a specific point in time.

The pathology representation also includes, amongst other things, references to *EKS* concepts representing type (e.g. 'Tumour') and location (e.g. 'Breast').

This example helps illustrate the two main types of procedures embodied within the *COM*. These are:

- **Dynamic *COM* expansion based on *EKS* knowledge, and *EKS*-related inference:** For example, the *EKS* that represents pathology-types will provide sets of 'descriptor' attributes for each concept (e.g. 'size' and 'shape' for the 'Tumour' concept). These attributes are used to dynamically create sets of both 'snapshot-descriptor' and 'history-descriptor' fields on the *PathologySnapshot* and *PathologyHistory* objects respectively. Furthermore, as the fields in the *COM* (both static and dynamically-created) are assigned values, the *EKS-related inference mechanisms* will be consulted, which may result in the provision of additional 'descriptor' attributes. Following on from the above example: if the 'location' field on the *PathologyHistory* object is assigned a value of 'Breast', then the inference mechanism will infer that an additional 'descriptor' attribute called 'herceptin-2-receptor' is required. This is a Boolean valued attribute that is only applicable to tumours located in the breast.

- **Dynamic data abstraction:** For example, the value of a 'history-descriptor' field is a dynamically updated temporal summary of the values of the corresponding set of 'snapshot-descriptor' fields (e.g. if the set of snapshots for 'Tumour' records 'size' as '2', '4' and '7' mm, the temporal summaries will include 'max-value = 7' and 'strictly-increasing = true').

The *COM* also involves the expression of various types of data/query creation constraint.

5. The CLEF Simulator

Testing or validation of the CLEF Chronicle is problematic because no real patient data contains the details required in machine readable form. Indeed, a major purpose of developing the chronicle is to guide efforts to improve patient records and make them more appropriate for research. Efforts to transcribe manually even small amounts of our experimental corpus of traditional electronic records proved so time consuming as to be impractical.

Our solution to this impasse has been to construct a simulator to model breast cancer patients. The processes modelled include the way in which tumour cell colonies grow, metastasise and cause local and systemic effects, the behaviour of the patient both as a biological system and as a sentient healthcare

consumer in response to local or systemic symptomatology arising from either the tumour or its treatment, and the actions of the clinician as a provider of investigations of varying sensitivity and treatments of variable effectiveness. The modelling process involves dynamic event-driven interactions through simulated time of all three actors (disease, patient and clinician). The simulator generates simulated clinical stories of similar content and complexity to real life, represented both as chronicles and as note-form text.

Whilst the simulator models were engineered to superficially approximate real populations of patients, they do not pretend to possess any predictive properties with respect to, for example, what effect any change in treatment efficacy, or clinical service reconfiguration, might have on real population outcomes. Rather, for our stated purpose of testing the Chronicle as a means to represent individual patient stories, it is sufficient to evaluate whether the complexity and content of individual generated clinical stories approximates that of real clinical stories: whether they have surface truth-likeness, or verisimilitude.

Judging the truth-likeness of generated stories is necessarily subjective and imprecise. Formal evaluation would be inherently problematic not least because of the difficulty in obtaining sufficient time from specialist clinicians to serve as the judges. At the present time, therefore, we base our claim that the simulated stories approximate real clinical stories on the following evidence:

- The prototype simulator was written by a clinician (JR) and is based on a detailed model of tumour, patient and clinician behaviour and their mutual dynamic interactions.

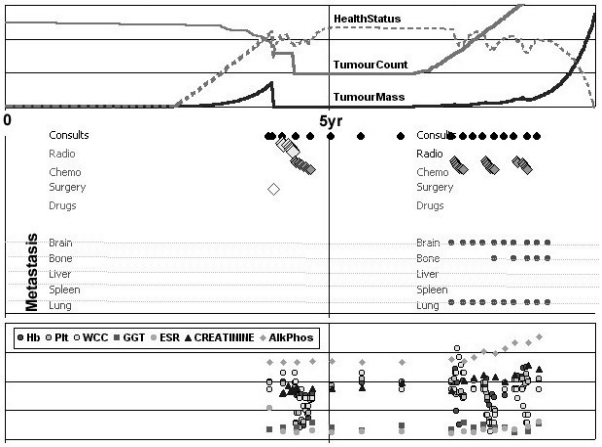


Figure 3 : graphical plot of simulator parameters over course of one simulation

- Inspection by JR of several hundred simulations, represented both as short note output (Figure 5) and graphical time-line views of key simulator parameters (Figure 3), did not identify any significantly implausible patients.
- Although not designed or required to produce simulated populations of patients with characteristics mimicking those of real populations, 10-year Kaplan-Meier curves (a standard tool for comparing cancer survival) for populations of simulated patients are similar to those for populations of real patients (Figure 4), including those curves representing subanalyses for tumour grade or disease stage at presentation.
- Two senior oncology physicians invited to comment on the generated narrative output of simulators suggested only minor changes (e.g. both AST and GGT should appear as liver damage markers).
- Queries similar to those forming the central research question in typical registered cancer trials may be constructed and executed against the simulated chronicle repository, as well as more detailed questions such as 'what effect on long term survival accrues from interrupting a course of chemotherapy in order to go on holiday'.

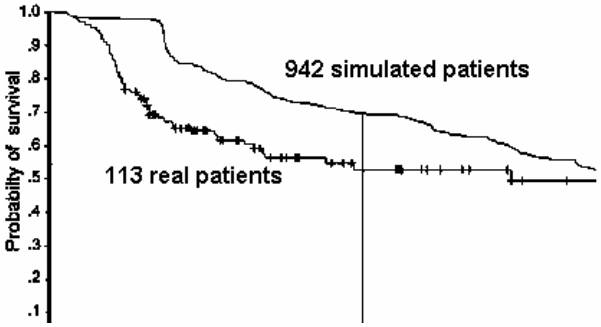


Figure 4 : Overall survival curves for real and simulated breast cancer patients

Whilst this provisional evaluation of the simulator suggests that its generated clinical stories are adequate simulacra of reality, its documentation of them in CLEF Chronicle format clearly significantly exceeds that of usual clinical practice both in detail and completeness: it includes many information holders that are not explicitly present in the traditional medical record, in particular the relationships between entities in the story. For example, every test has an explicit causal relation with the phenomenon (usually a problem) that was the reason for requesting the test. Additionally, the simulator records clinically significant entities that are often not only entirely absent from the traditional clinical record, but also would be difficult to automatically infer. These include significant negative investigation findings, the stage of the disease at a given point in time, or exactly when

(and on what evidence) a patient changed status from being in remission to being in relapse.

```
Patient notices a problem aged 46 (wk 209)
** Patient presents aged 45.8 (wk 212) **
Ix:Xray chest: normal
Ix:Clinical examination showed: a new 4.06cm breast mass
Follow-up in 3 weeks with results to plan treatment
** Initial Treatment Consult: week 215 **
** Surgery **
4.26 cm lesion was incompletely excised
Ix:Histopathology report on excision biopsy
Grade 8 invasive ductal adenocarcinoma of breast.
Oestrogen receptor negative
21 nodes were positive
Six weeks of radiotherapy to axilla should follow surgery
Secondaries: commence 6 weeks of radiotherapy.
Secondaries: commence 6 weeks of systemic chemotherapy.
High grade tumour: for chemotherapy
Tumour is hormone insensitive. No hormone antagonist
Follow-up in 8 weeks
** Consult to assess response to treatment : week 223 **
Ix:Xray chest: normal
Ix:Clinical examination normal
No evidence of further tumour.
See in 8 weeks
Radiotherapy cycle given. 5 remaining
Radiotherapy cycle 5 deferred because patient on holiday
Radiotherapy cycle given. 4 remaining
                    <<< SNIP >>>
 ** Consult to assess response to treatment : week 428 **
Ix:Xray chest: abnormal.
No new lesions found
2 old lesions found
Largest measures 1cm, smallest is 0.7cm
High alkaline phosphatatse: for bone scan
Ix:Bone Scan: abnormal.
No new lesions found
17 old lesions found
Largest measures 1.2cm, smallest is 0.7cm
Patient confused: for CTScan Brain
Ix:CTScan of brain: abnormal.
No new lesions found
6 old lesions found
Largest measures 1.4cm, smallest is 0.8cm
High grade unresponsive tumour. Palliative Care
See in 8 weeks
** Died aged 50 from cancer (week 475)
.....
Grade 8 - 23 local mets, 70 distant metastasis
Seen in clinic 19 times and 4 course of chemo prescribed
Total tumour mass 1374700 in 200 tumours of which 48
detected.
ESR=160 Creatinine=559
41 tumours were destroyed or removed
54 bony mets, total mass 740941 d=5.41cm AlkPhos=4811
```

Figure 5: Extract of simulator 'short note' output

Using the simulator output, we may cautiously estimate the richness and complexity that 'real' clinical stories – and aggregations of them - might take on if represented as Chronicles. An analysis of a simulated population of 986 patient chronicles revealed:

- Individual patient chronicles comprise an average of 385 object instances and 753 semantic relations
- The most complicated patient story required 2295 instances and 4636 relationships
- The aggregated repository, comprising 986 discrete networks, contained 382,103 instances and 742,595 semantic relations.

The prototype simulator and documentation is available for download from:
http://www.clinical-escience.org/simulator.html
Subsequent releases of the reimplementation may be made available through the same URL in future.

We have now developed a more generalised Java re-implementation of the simulator that is capable of creating richer patient records, in a format suitable for populating the *Chronicle Object Model*.

6. Future Challenges

Outstanding issues concerning the *Chronicle Representation* include:

- How to store very large numbers of *Chronicle Object Model* instances persistently such that queries may be constructed, and efficiently executed, over aggregations of such instances.
- How to implement such a query mechanism involving both ontological and temporal reasoning.
- How to deal with data arising from the 'chroniclisation' of real records will almost certainly be both incomplete and 'fuzzy'.

Other issues concern the presentation of the chronicle to the clinician, including:

- How the clinical content of a reconstructed chronicle can be visualised in order to validate its content against the more traditional patient record data from which it has been derived.
- How complex queries over sets of patient chronicles can best be formulated by the ordinary clinician.

7. Discussion

The idea of representing clinical information as some form of semantic net, particularly focussing on why things were done, is not new: echoes of it can be found in Weed's work on the problem oriented record [7]. Ceusters and Smith more recently advocated the

resolution of coreferences in clinical records to instance unique identifiers (IUIs) [8].

The semantic web initiative offers new possibilities for implementing such an approach, but the lack of any suitable clinical data severely constrains any practical experimentation. The CLEF Simulator provides a useful means to explore some of the computational and representational issues that arise.

8. References

1. Taweel A, Rector, AL, Rogers J, Ingram D, Kalra D, Gaizauskas R, Hepple M, Milan J, Power R, Scott D, Singleton P. (2004) CLEF – Joining up Healthcare with Clinical and Post-Genomic Research. *Current Perspectives in Healthcare Computing*:203-211
2. Harkema H, Roberts I, Gaisauskas R, Hepple M. (2005) A web service for biomedical term look-up. *Comparative and Functional Genomics* 6;1-2:86-83
3. Rector A, Nowlan W, Kay S. (1991) Foundations for an Electronic Medical Record. *Methods of Information in Medicine*;30:179-86.
4. Beale T. (2003) Archetypes and the EHR. *Stud Health Technol Inform.* 2003;96:238-44
5. Grenon P, Smith B (2004) SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition and Computation* 4;1:69-104
6. Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P. (2003) The Description Logic Handbook. Cambridge University Press. ISBN: 0521781760
7. Weed LI (1969) Medical records medical education, and patient care. The problem-oriented record as a basic tool. Cleveland, OH: Case Western Reserve University
8. Ceusters W, Smith B. (2005) Strategies for referent tracking in Electronic Health Records. *Journal of Biomedical Informatics* (*in press).*
9. Tsarkov D, Horrocks I. (2006) FaCT++ Description Logic Reasoner: System Description. *Proc of Int. Joint Conf. on Automated Reasoning (IJCAR~2006) (in press).*

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

The BIOPATTERN Grid – Implementation and Applications

P. Hu [1], L. Sun [1], C. Goh [1], B. Hamadicharef [1], E. Ifeachor [1,a], I. Barbounakis [2], M. Zervakis [2], N. Nurminen [3], A. Varri [3], R. Fontanelli [4], S. Di Bona [4], D. Guerri [4], S. La Manna [4], K. Cerbioni [5], E. Palanca [5] and A. Starita [5]

[1] School of Computing, Comm. and Electronics, University of Plymouth, UK
[2] Telecommunications System Institute, Technical University of Crete, Greece
[3] Institute of Signal Processing, Tampere University of Technology, Finland
[4] Synapsis S.r.l. in Computer Science, Italy
[5] Computer Science Department, University of Pisa, Italy

Abstract

The primary aim of this paper is to report the development of a new testbed, the BIOPATTERN Grid, which aims to facilitate secure and seamless sharing of geographically distributed bioprofile databases and to support analysis of biopatterns and bioprofiles to combat major diseases such as brain diseases and cancer within a major EU project, BIOPATTERN (www.biopattern.org). The main objectives of this paper are 1) to report the development of the BIOPATTERN Grid prototype and implementation of BIOPATTERN grid services (e.g. data query/update and data analysis for brain diseases); 2) to illustrate how the BIOPATTERN Grid could be used for biopattern analysis and bioprofiling for early detection of dementia and for assessment of brain injury on an individual basis. We highlight important issues that would arise from the mobility of citizens in the EU and demonstrate how grid can play a role in personalised healthcare by allowing sharing of resources and expertise to improve the quality of care.

1. Introduction

Grid computing is aimed to provide a global Information Communication Technology (ICT) infrastructure to facilitate seamless and secure sharing of geographically distributed resources (e.g. data, storage, computation, algorithms, applications and networks). Great efforts, resources and funding have been put into national, regional and international initiatives in grid infrastructure, grid core technologies and grid applications. The integration of grid computing and healthcare has formed a new exciting and specialist area called Healthgrid. Examples of healthcare applications include distributed mammography data retrieval and processing (e.g. the EU's MammoGrid [1] and the UK's eDiaMoND [2] projects), and multi-centre neuro-imaging (e.g. the USA's BIRN [3] and Japan's BioGrid [4]). There is a trend in modern medicine towards individualisation of healthcare and, potentially, grid computing can also play a role in this by allowing sharing of resources and expertise to improve the quality of care.

In this paper, we report efforts to exploit grid computing to support individualisation of healthcare to combat major diseases such as brain diseases within a major EU-funded, Network of Excellence (NoE) project, BIOPATTERN (www.biopattern.org). The Grand Vision of the project is to develop a pan-European, coherent and intelligent analysis of a citizen's bioprofile; to make the analysis of this bioprofile remotely accessible to patients and clinicians; and to exploit bioprofiles to combat major diseases such as cancer and brain diseases. A biopattern is the basic information (pattern) that provides clues about underlying clinical evidence for diagnosis and treatment of diseases. Typically, it is derived from specific

[a] Corresponding Author: Prof Emmanuel Ifeachor, School of Computing, Communications and Electronics, University of Plymouth, Plymouth PL4 8AA, U.K. Email: E.IFeachor@plymouth.ac.uk.

data types, e.g. genomics and proteomic information and biosignals, such as the electroencephalogram (EEG) and Magnetic Resonance Imaging (MRI). A bioprofile is a personal `fingerprint' that fuses together a person's current and past medical history, biopatterns and prognosis. It combines data, analysis, and predictions of possible susceptibility to diseases. It will drive individualisation of care.

The aim of the BIOPATTERN Grid is to facilitate secure and seamless sharing of geographically distributed bioprofile databases and to support analysis of biopatterns and bioprofiles to combat major diseases such as brain diseases and cancer.

The main objectives in this paper are 1) to report the development of a new Grid test bed, the BIOPATTERN Grid, for biopattern analysis and bioprofiling in support of individualisation of healthcare. We focus on the implementation of the BIOPATTERN Grid prototype and the development of grid services; 2) to illustrate the applications of the BIOPATTERN Grid, we present two pilot applications – use of the BIOPATTERN Grid for early detection of dementia and for assessment of brain injury.

The remainder of the paper is organised as follows. In Section 2, the BIOPATTERN Grid architecture and prototype are described. In Section 3, the detailed implementations of BIOPATTERN Grid services are presented. In Section 4, two pilot applications of BIOPATTERN Grid are illustrated. Section 5 concludes the paper.

2. BIOPATTERN Grid Architecture and Prototype

The architecture of BIOPATTERN Grid is divided in four layers as shown in Figure 1. The Grid Portal serves as an interface between an end user (e.g. a clinician) and the BIOPATTERN Grid network. The Grid services layer provides advanced services for data acquisition (including remote automatic data acquisition), data analysis & visualisation for brain diseases and cancer, and data query and/or information crawling services. The Grid middleware provides grid functionalities for security, resource management, information service, data management and data services support. The Globus Toolkit 4 (GT4) [5] is chosen to implement Grid middleware functions. Condor [6] is used for job queuing, job scheduling and to provide high throughput computing. The grid resources layer, contains computational resources, data resources, and knowledge resources (e.g. algorithms pool) and networks.

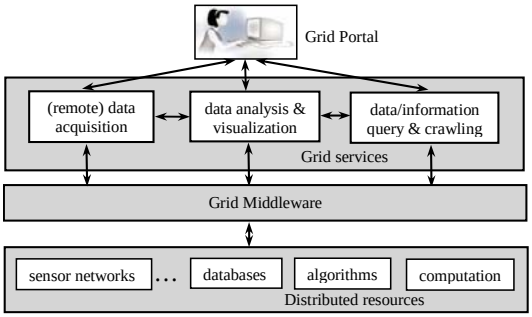


Figure 1. BIOPATTERN Grid Architecture

We have built a BIOPATTERN Grid Prototype within the BIOPATTERN Consortium to provide a platform for clinicians and researchers to share information in distributed bioprofile databases and computational resources. A major aim of the prototype is to facilitate analysis, diagnosis, and care for brain diseases and cancer. Currently, the prototype connects five sites –the University of Plymouth (UOP), UK; the Telecommunication System Institute (TSI), Technical University of Crete, Greece; the University of Pisa (UNIPI), Italy; Synapsis S.r.l. (Synapsis), Italy, and Tampere University of Technology (TUT), Finland (see Figure 2). Each site may hold bioprofile databases, Grid nodes, Condor pool, high performance cluster, algorithms pool, Grid portal, or an interface to remote data acquisition networks. For example, at University of Plymouth (UoP) node, it contains an algorithm pool including key algorithms for brain diseases analysis (e.g. fractal dimension algorithm for early detection of dementia and Independent Component Analysis based algorithm for assessment of brain injury); bioprofile databases which contain basic patient's clinical information, EEG data (awake EEG at resting state) for dementia, and EEG data for brain injuries); a web server which holds the BIOPATTERN Grid Portal; a condor pool, named PlymGRID, which contains up to 1400 nodes within the Plymouth campus.

3. Implementation of BIOPATTERN Grid Services

The BIOPATTERN Grid provides both high level and low level services. The high level services are implemented mainly for end users (e.g. clinicians or researchers), who have permissions to use specified grid-enabled services (e.g. to access distributed resources or to compare results from existing algorithms) without any grid knowledge (e.g. underlying grid techniques, locations of resources and detailed access constrains to resources). The low level services are implemented and provided mainly for grid services/applications developers to access BIOPATTERN Grid services directly, or to develop interfaces with BIOPATTERN Grid (e.g. the current AmI-Grid project within BIOPATTERN is seeking the integration of AmI's wireless data acquisition system with BIOPATTERN Grid).

3.1 High Level Services

Three high level services have been implemented in the BIOPATTERN Grid. As shown in Figure 3, the services are clinical information query, clinical information update and EEG analysis. The BIOPATTERN Grid Portal serves as the interface between an end user (e.g. a clinician or a researcher) to the BIOPATTERN Grid. It allows an end user to access these services from any where with Internet access (via a web browser).

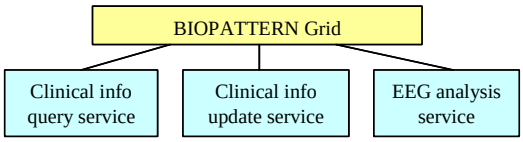


Figure 3. High level services provided by the BIOPATTERN Grid

3.1.1 Clinical Information Query Service

This service is designed based on a scenario that an end-user (e.g. a clinician) wants to query the information of a patient and to download the patient's EEG data for off-line analysis (the patient's information and EEG data are all distributed across the BIOPATTERN Grid network). To achieve this, the end-user has to go through certain steps via the BIOPATTERN Grid Portal, e.g. 1) Query for a specified patient; 2) Select EEG data files of the patient to download.

The query service is implemented in four levels from the top customer GUI to the bottom grid resources level as shown in Figure 4.

The customer GUI level provides interfaces to allow end-users to seamlessly access grid-enabled services with graphic visions. The query interface and file download interface are designed to enable users to access required clinical information without any grid knowledge, and to allow users to simply follow the defined steps to use this service. The functionalities provided at this level include handling HTTPS requests (e.g. obtainment of requests from end-users by filling e-forms), forwarding such requests to the lower level, etc. JSP and html are languages used to implement these interfaces, which are held by Tomcat containers.

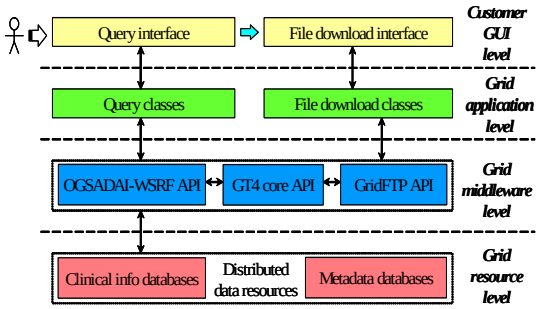


Figure 4. Implementation of clinical information query service

For the grid application level, two types of classes: query classes and file download classes, are responsible to translate specified requests from the interfaces in the customer GUI level into detailed action requisitions and further send those requisitions to grid servers for execution. The query classes can provide concurrent access of distributed databases via OGSADAI-WSRF in order to obtain information as descriptions of specified patients, EEG data locations, etc. The file download classes mainly offer file transfers from the grid nodes which contain specified EEG data to the grid nodes which can be accessed by end-users.

The grid middleware level contains various APIs, including OGSADAI-WSRF API, GT4 core API, GridFTP API, etc. All these APIs are responsible for dealing with action requisitions that come from the grid application level and realising such requisitions by accessing distributed data resources.

The grid resource level holds distributed data resources, which are described as low level

services held by Globus containers. Data resources that support the query service includes clinical information databases (e.g. database for personal information of patients and EEG data), and metadata databases (e.g. database for descriptions of EEG data).

3.1.2 Clinical Information Update Service

This service is designed based on a scenario that an end-user (e.g. a clinician) needs to update an existing patient's clinical records (e.g. a new EEG recording has been taken and needs to be uploaded to the distributed bioprofile databases or a new patient's clinical information has to be added to the bioprofile databases). To achieve this, the end-user needs to go through the following steps via the Portal, e.g. 1) Add a new patient and his/her information, or query for an existing patient; 2) Upload EEG data files of the patient.

Similar to the query service, the update service is also implemented in four levels, as shown in Figure 5.

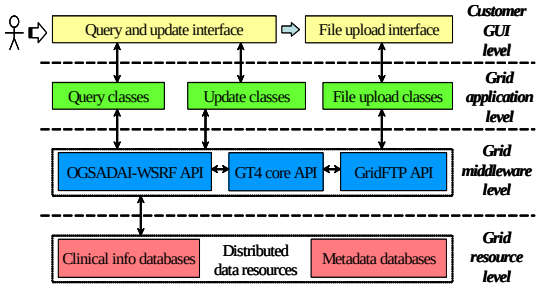


Figure 5. Implementation of clinical information update service

For this service, we have implemented both query and update interfaces in the customer GUI level in order to allow end-users to be able to easily insert information of new patients to the databases or to find a specified patient to update his/her records.

3.1.3 EEG Analysis Service

This service is designed based on a scenario that an end-user (e.g. a clinician) wants to use automated analysis services provided by BIOPATTERN Grid to help for the detection/diagnosis of dementia or to help for the assessment of brain injury (two services are provided currently).

To achieve this, end-users have to go through several steps via the portal, e.g. 1). select for analysis either for dementia or for brain injury; 2) query for a specified patient; 3) select the patient's EEG data files for analysis and submit analysis jobs; 4) choose the way of viewing analysis results (e.g. in canonograms and/or bargraphs for dementia).

The EEG analysis service is also implemented in four levels, as presented in Figure 6.

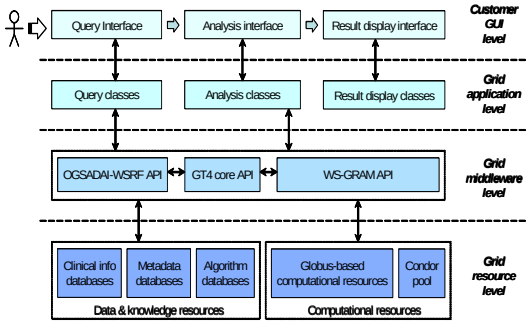


Figure 6. Implementation of EEG analysis services

In this service, the customer GUI level provides three user friendly interfaces, as query, analysis and result display interfaces, which are connected with their corresponding types of Java classes at the grid application level, as presented in Figure 6. For those Java classes, the analysis classes mainly offer generation of grid jobs (e.g. generation of Resource Specification Language files for job description), finding appropriate computational grid resources (e.g. looking for light-loaded resources based on job requirements and resource information), and submission of jobs, retrieving analysis results and forwarding the results to the high level for display. The results display classes enable end-users to view such analysis results in graphical display. The direct connection between result display classes and grid middleware level will be established in the future in order to support more complex functionalities, such as 3D visualisation.

Different to the other two high level services, the EEG analysis service holds both computational and data resources. The computational resources cover both High Throughput Computing (HTC) and High Performance Computing (HPC) resources. The Condor pool, currently plays a key role in HTC-based EEG analysis. The Globus-based computational resources support the HPC-based EEG analysis at present.

3.2 Low Level Services

There are two types of low level services which have been considered in the implementation, BIOPATTERN data services and computational services.

3.2.1 BIOPATTERN Data Services

The goal of the provision of the data services is to support certain features in clinical data access and integration over distributed heterogeneous data resources, such as database federation and data transformation.

Thus far we have mainly investigated into the area of database federation and implemented some basic data services, such as the generic query and update services, based on OGSADAI-WSRF, an OGSADAI distribution compliant with WSRF specifications.

Figure 7 presents a three-layer architecture for the implementation. In the architecture, the top layer houses all BIOPATTERN data services, which can provide functionalities, such as data access and transformation to permitted and authorized grid users based on specific requirements. The middle layer that holds the OGSADAI grid data service acts as an interface between the top and bottom layers to make the underlying data sources be accessible to those grid users.

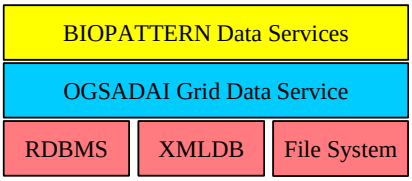


Figure 7. Low level data services architecture

The design of this architecture enables grid users to access different types of data resources within the BIOPATTERN Grid in a consistent and resource-independent way and also offers the flexibility for further development in areas like data integration, security, access constrains, etc.

In the implementation, both BIOPATTERN data services and OGSADAI grid data service are built as Web services and held by Globus containers.

3.2.2 BIOPATTERN Computational Services

The computational services are designed to support those computation intensive applications, such as medical image analysis and protein sequence comparison, and mainly implemented based on WS-GRAM.

On any BIOPATTERN grid nodes, a grid user can use this type of services to submit jobs to any permitted HPC and/or HTC resources and then get expected results back. The completion of a grid job may require several steps, i.e. 1) creation of the job; 2) stage in (e.g. transmission of input files); 3) execution of the job; 4) stage out (e.g. transmission of output files); 5) job cleanup.

Currently, all computational services are held by Globus containers located at certain grid nodes (e.g. nodes at UoP and TSI) in the BIOPATTERN Grid.

4. Applications of the BIOPATTERN Grid

In line with the Grand Vision of the BIOPATTERN project, the applications of the BIOPATTERN Grid are sought mainly around the areas of brain diseases and cancer. Currently we have developed two applications on early detection of dementia and assessment of brain injury. These applications are accessible via the BIOPATTERN Grid Portal and are developed for proof of the concept. It will be used as a vehicle for future clinical validation.

4.1 BIOPATTERN Grid for Early Detection of Dementia

Dementia is a neurodegenerative cognitive disorder that affects mainly elderly people. Several objective methods are available that may support early diagnosis of dementia. Among others, the EEG which measures electrical activities of the brains is regarded as an acceptable and affordable method in the routine screening of dementia in the early stages. Different biodata analysis methods (e.g. Fractal Dimension or FD) are developed which try to provide a biomarker (e.g. FD index) to indicate possible on-set of dementia. Using current clinical criteria, delay between the actual onset and clinical diagnosis of dementia is typically 3 to 5 years. Serial EEG recordings

are normally suggested for elderly people in order to diagnosis dementia at an early stage [7].

Due to mobility of a citizen, a patient's serial EEG recordings can be located in different clinical centres and possibly across different countries. We envisage that grid computing can play a role in early detection of dementia when dealing with geographically distributed data resources.

To illustrate the concept of BIOPATTERN Grid for early detection of dementia, a hypothetical patient pool consisting of 400 subjects, each with three EEG recordings was created. These data are hypothetical representation of recordings taken at three time instances akin to longitudinal studies carried out in reality. The datasets are distributed at TSI, TUT and UoP sites. The FD analysis algorithm is used to compute the FD of each dataset.

Through the Portal, an end-user (e.g. a clinician) can select a patient, e.g. Mike, and the algorithm is used to perform the analysis. Upon submission, Mike's information, including his serial EEG recordings (located in UoP, TSI and TUT, respectively) are retrieved and analyzed. Results can be shown in canonograms (see Figure 8) where changes in the EEG indicating Mike's conditions. The canonograms (from left to right) show the FD values of the Mike's EEG taken at time instances of 1 (data at TSI), 2 (data at TUT) and 3 (data at UoP) respectively. The FD values for the left canonogram indicates Mike in a normal condition with high brain activity, whereas the FD value for the right canonogram indicates Mike in a probable Alzheimer Disease with low brain activity. The middle one shows the stage in between. Changes in the FD values provide some indication about the disease progression. This can help clinicians to detect dementia at an early stage, to monitor its progression and response to treatment.

4.2 BIOPATTERN Grid for assessment of brain injury

The second application of the BIOPATTERN Grid is used to assess the severity of brain injury by analysing evoked potentials (EPs) buried in raw EEG recordings. The assessment process consists of two stages: 1) an Independent Component Analysis (ICA) [8] algorithm is used to extract single trial evoked potential activity of clinical interest and discard irrelevant components such as background EEG and artefacts; 2) a LORETA (LOw Resolution Electromagnetic TomograPhy) [9] is used to localise the source of brain activity. These methods are implemented in Matlab and compiled as executable (using the Matlab run-time library) and distributed over the Condor pool to speed up the time-consuming analysis. Figure 9 shows an example of results of such analysis via the portal, with topography maps (only for the two first ICA components) of one normal patient (top) and one patient (bottom). The head model, which is shown, is assumed to be an 8cm sphere and is represented with 8 slices (the lower slice is the left and the higher is the right). Through this service, a clinician can easily use the advanced EEG analysis algorithm (e.g. ICA-LORETA) to analyze a patient's raw EEG data for assessing brain injury.

5. Conclusion

In this paper, we have presented a new testbed, BIOPATTERN Grid which aims to share geographically distributed computation, data and knowledge resources for combating major diseases (e.g. brain diseases). We illustrated the development of BIOPATTERN Grid Prototype, the implementation of the BIOPATTERN Grid services and two pilot applications for early detection of dementia and for assessment of brain injury.

BIOPATTERN Grid is an ongoing project and results presented here are limited in scale. In the near future, the BIOPATTERN Grid prototype will be extended to include more grid nodes (both on Globus and on Condor), more computing resources (e.g. connecting with HPC clusters in UNIPI), more grid applications (e.g. to include cancer diagnosis and prognosis) and services (e.g. data transformation), integration with other projects within the BIOPATTERN, such as to integrate with AmI-Grid project [10] for connection with wireless data acquisition networks and to integrate with grid.it [11][12] to provide crawling services for information query. In the long term, we also seek the integration of the BIOPATTERN Grid with other regional or national Grid projects/networks (e.g. EU's EGEE).

Due to the nature of healthcare, the BIOPATTERN Grid will need to address several issues such as regulatory, ethical, legal,

privacy, security, and QoS, before it can move from research prototype to actual clinical tool.

Acknowledgement

The authors would also like to thank Dr. C. Bigan from EUB, Romania for providing the original EEG data sets and Dr. G. Henderson for providing the algorithms for computing the FD index. We acknowledge the financial support of the European Commission (The BIOPATTERN Project, Contract No. 508803) for part of this work.

References

[1]. S. R. Amendolia, F. Estrella, C. D. Frate, J. Galvez, W. Hassan, T Hauer, D Manset, R McClatchey, M Odeh, D Rogulin, T Solomonides and R Warren, "Development of a Grid-based Medical Imaging Application", Proceedings of Healthgrid 2005, from Grid to Healthgrid, 2005, pp.59-69.

[2]. S. Lloyd, M. Jirotka, A. C. Simpson, R. P. Highnam, D. J. Gavaghan, D. Watson and J. M. Brady, "Digital mammography: a world without film?", Methods of Information in Medicine, Vol.44, No. 2, pp. 168-169, 2005.

[3]. J. S. Grethe, C. Baru, A. Gupta, M. James, B. Ludaescher, M. E. Martone, P. M. Papadopoulos, S. T. Peltier, A. Rajasekar, S. Santini, "Biomedical Informatics Research Network: Building a National Collaboratory to Hasten the Derivation of New Understanding and Treatment of Disease", Proceedings of Healthgrid 2005, from Grid to Healthgrid, 2005, pp. 100-109.

[4]. K. Ichikawa, S. Date, Y. Mizuno-Mastumoto, and S. Shimojo, "A Grid-enabled System for analysis of Brain Function", In Proceedings of CCGrid 2003 (3rd IEEE/ACM International Symposium on Cluster Computing and the Grid), May 2003.

[5]. Foster, "Globus Toolkit Version 4: Software for Service-Oriented Systems", Proceedings of IFIP International Conference on Network and Parallel Computing, 2005, pp. 2-13.

[6]. http://www.cs.wisc.edu/condor/hawkeye

[7]. G. T. Henderson, E. C. Ifeachor, H. S. K. Wimalartna, E. Allen and N. R. Hudson, "Prospects for routine detection of dementia using the fractal dimension of the human electroencephalogram", MEDSIP00, pp. 284-289, 2000.

[8]. T-W Lee, M. Girolami, TJ. Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed sub-Gaussian and super-Gaussian sources". Neural Computation 1999;11(2):606-633.

[9]. R. D. Pascual-Marqui. "Review of methods for solving the EEG inverse problem" International Journal of Bioelectromagnetism 1999, 1: 75-86

[10]. M. Lettere, D. Guerri, R. Fontanelli. "Prototypal Ambient Intelligence Framework for Assessment of Food Quality and Safety", 9th Int. Congress of the Italian Association for Artificial Intelligence (AI*IA 2005) – Advances in artificial Intelligence, pp. 442-453, Milan (Italy), Sep. 21 - 23, 2005

[11]. Grid.it: "Enabling Platforms for High-Performance Computational Grids Oriented to Scalable Virtual Organizations", http://grid.it/.

[12]. K. Cerbioni, E. Palanca, A. Starita, F. Costa, P. Frasconi, "A Grid Focused Community Crawling Architecture for Medical Information Retrieval Services", 2nd Int. Conf. on Computational Intelligence in Medicine and Healthcare, CIMED'2005.

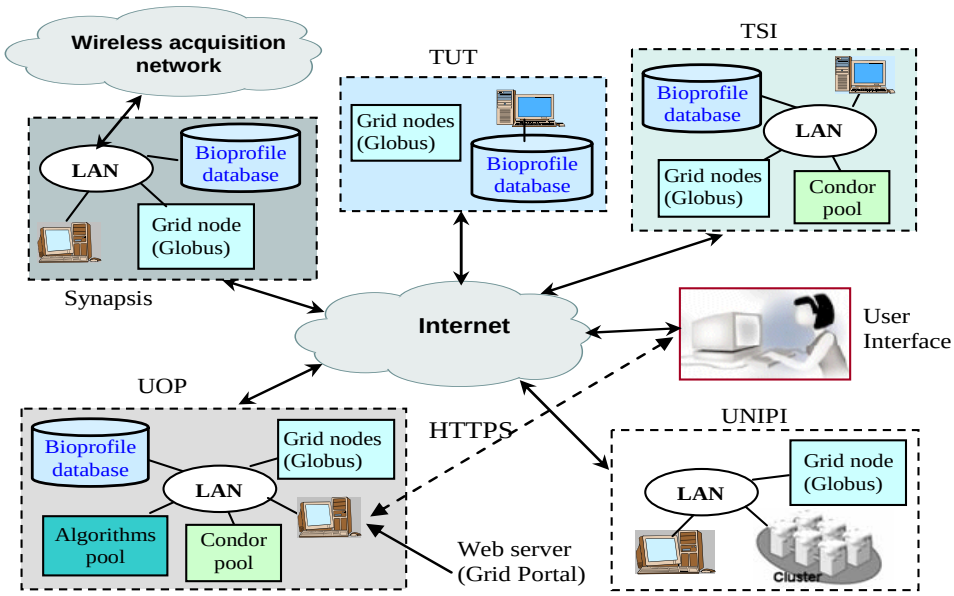


Figure 2. BIOPATTERN Grid Prototype

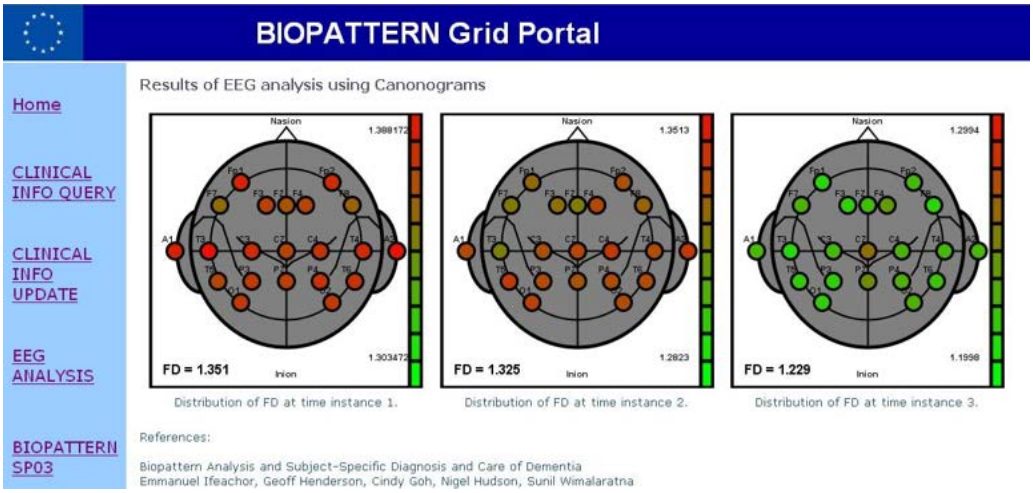


Figure 8. Canonograms showing the distribution of FD values for EEG analysis for dementia

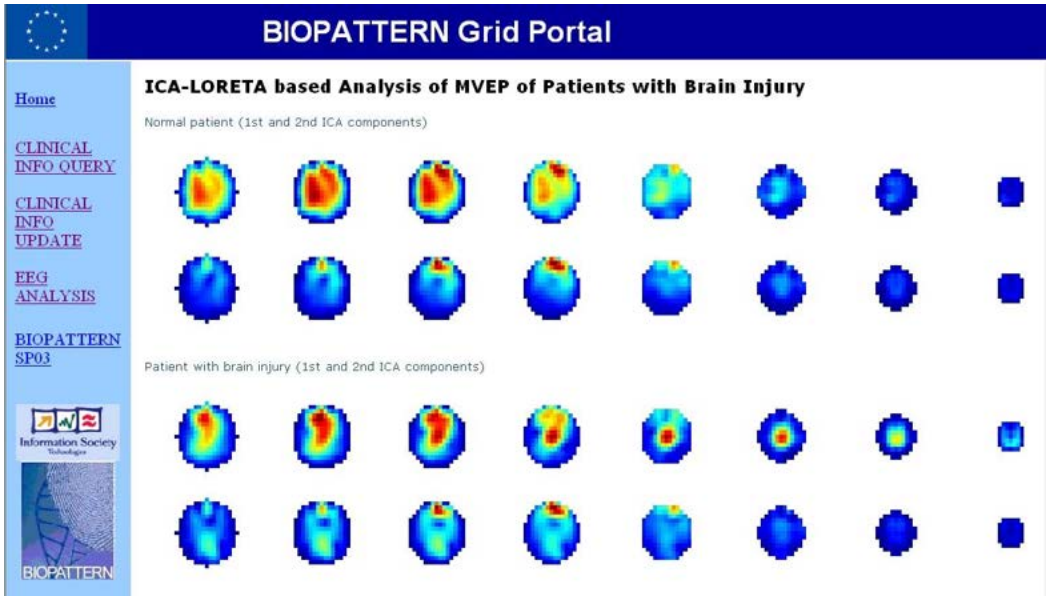


Figure 9. Topography maps of normal subjects and patients with brain injury

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Using *e*Science to calibrate our tools: parameterisation of quantum mechanical calculations with grid technologies

K. F. Austen[1], T. O. H. White[1], R. P. Bruin[1], M. T. Dove[1], E. Artacho[1], R. P. Tyer[2]
[1]Department of Earth Sciences, University of Cambridge, Downing Street, Cambridge, CB2 3EQ
[2]CCLRC Daresbury Laboratory, Daresbury, Warrington, Cheshire, WA4 4AD

Abstract

A report is presented on the use of *e*Science tools to parameterise a quantum mechanical model of an environmentally important organic molecule. *e*Science tools are shown to enable better model parameterisation by facilitating broad parameter sweeps that would otherwise, were more conventional methods used, be prohibitive in both time required to set up, submit and evaluate the calculations, and in the volume of data storage required. In this case, the broad parameter sweeps performed highlighted the existence of a computational artefact that was not expected affect this system to such an extent, and which is unlikely to have been observed had fewer data points been taken. The better parameterisation of the model leads to more accurate results and the better identification of the applicability of aspects of the model to the system, such that great confidence can be put in the results of the research, which is of environmental importance.

1. Introduction

Polychlorinated biphenyls (PCBs) have long been known to have environmental significance due to their persistence within the environment, and their high toxicity. New information on the interaction of these chemicals with common soil minerals is continually being sought to facilitate meso-scale modelling of their movement through aquifers, through assessment of the retardation effects of different minerals and their adsorption isotherms.

The structure of biphenyl is shown in Figure 1. PCBs share this structure, and each hydrogen atom on the carbon rings can be substituted for chlorine atoms, in any combination. The number of possible PCB congeners, 209 different arrangements of chlorine atoms around the biphenyl rings, poses an arduous and time-consuming problem for the typical computational scientist. Were all of these congeners to be studied by hand, the time required to generate the starting structures alone would be prohibitive, without even considering the best parameterisation of the model.

This paper reports the work that has been carried out, using *e*Science tools, to refine the input parameters for the PCB calculations. A large number of multi-dimensional parameter sweeps have been performed; and through the generation of large data sets, the importance of further parameterisation has been realised, which previously might have gone unnoticed. As a consequence, the accuracy of the results is optimised beyond that reached *via* conventional methods.

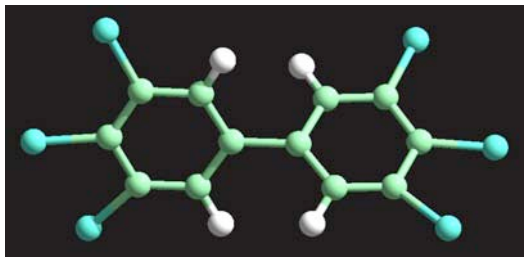


Figure 1 PCB structure (Cl=blue, C=green, H=white)

Investigations have already been made into the adsorption of the related molecules, polychlorinated dibenzodioxins (PCDDs) and polychlorinated dibenzofurans (PCDFs) [1], of which there are, respectively, 76, and 126, different congeners, onto the (001) surface of the clay mineral, pyrophyllite. The *e*Science tools, developed within the *e*Minerals project, generate each congener at various heights above the surface, and allow for relaxation of the geometries subject to certain constraints on the system [1]. The same tools will be used to study PCBs at the surface, but an additional complication occurs when investigating PCBs, as there is the possibility of rotation of the phenyl rings around the C–C bond that joins them. The ease of this rotation is expected to greatly influence the adsorption energies of these molecules onto the surface, and consequently it is extremely important that it is adequately described in the calculations. To this end,

an investigation has been made of the applicability of the current model to quantify of the energetics associated with the 360˚ rotation of the (Cl–)C–C–C–C(–Cl) torsion angle for 2,2'-dichloro biphenyl (hereafter 2-PCB). There has been work in the literature both experimentally [2] and computationally [3], which has shown that previous calculations of the groundstate equilibrium angle have not been able to reproduce the 75˚ angle for the molecule [3].

The role that *e*Science played in this work was integral to the methodology. The many tools that have been developed within the *e*Minerals project have been indispensable in enabling the individual scientist to use to quickly begin work and to perform initial, very detailed, exploration of the system, in a way that would not have otherwise been possible.

2. Methodology

2.1 Simulation Details

All the calculations reported here were carried out at the density functional theory (DFT) level using the latest version of the SIESTA code[4], which includes CML output and z-matrix input. In the first instance, the calculations have been performed using the auto-generated double zeta polarized (DZP) basis sets within the code and the PBE functional[5].

The study has taken place in two parts. Initially, the box size surrounding an hexa–chlorinated PCB molecule was converged with respect to the total energy of the system. The aim of this was to determine the minimum box size necessary to surround the molecule without any self-interaction between periodic images. This is useful for two reasons: first, a smaller box size means less computational expense; second, the box size will determine the lower-limit of the surface size necessary when the interaction of the molecules with the surface is investigated.

The first part of the study was performed with the 3,4,5,3',4',5'-hexachloro biphenyl molecule (6-PCB), for reasons explained below.

Once the minimum box size was determined, a starting structure for the 2,2'-dichloro biphenyl molecule was generated and then described using the z-matrix format within the SIESTA code [4]. The z-matrix format allows the constraint of parameters within the molecule, such as, in this case, the torsion angle between the two phenyl rings (Figure 2). The torsion angle was varied over 360˚ at 5˚ intervals, requiring 72 calculations. Two basis sets were tested; the auto-generated DZP SIESTA basis set and a user-specified basis set, the latter of which was the more computationally expensive of the two.

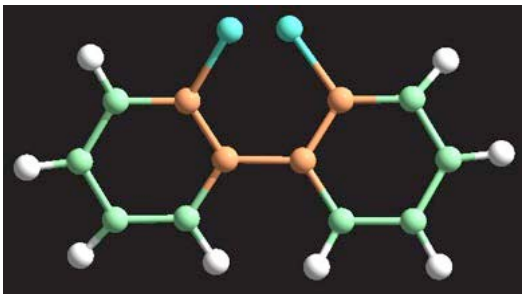


Figure 2 2,2'-dichloro biphenyl showing definition of torsion angle (red), the angle between the two planes defined by the pairs of carbons highlighted on each phenyl ring

2.2 Obtaining Starting Configurations

Starting configurations were required for both of the PCB molecules and these were obtained from experimental crystal structures stored in the Cambridge Crystallographic Database [6]. As the vacuum structure will differ from the crystal structure, further structural relaxation was required.

2.3 Calculating Box Size

Modelling of the molecule in vacuum was carried out using Periodic Boundary Conditions. The molecule is enclosed in a virtual box, the dimensions of which were varied to find the optimal size. The PCB chosen for the box size calculation was, as mentioned above, the 3,4,5,3',4',5'-hexachloro biphenyl (6-PCB) molecule. The chlorine atoms are larger, and the C–Cl bond lengths longer, than is the case for hydrogen. The fully chlorinated PCB molecule, where all 8 hydrogen atoms are replaced by chlorine atoms, posed difficulties in calculating the fixed planar geometry due to unfavourable steric repulsions between chlorine atoms in the 2 and 6 positions on opposite rings. 6-PCB was chosen, therefore, because it has the largest space-filling contributions of all the possible PCBs that can easily be calculated in a planar configuration. The molecule has the approximate dimensions of 10Å in length and 5Å across the chlorophenyl rings (Figure 1).

As such, an initial box size was taken to be 15Å x 10 Å x 10Å, and the molecule aligned so that it lay lengthwise along the long axis of the cuboid. The two shorter box–sides were kept equal to allow for free rotation around the C–C bond without interference between periodic images. The side lengths were sequentially increased up to 25 Å for the c parameter and 20 Å for a and b, at 0.5 Å intervals. This required the calculation of over 100 box sizes, each with different values of the lattice parameters. The generation and management of these calculations is detailed in Section 2.5.

The broad parameter sweeps, enabled by the use of *e*Minerals submission scripts, allowed such a

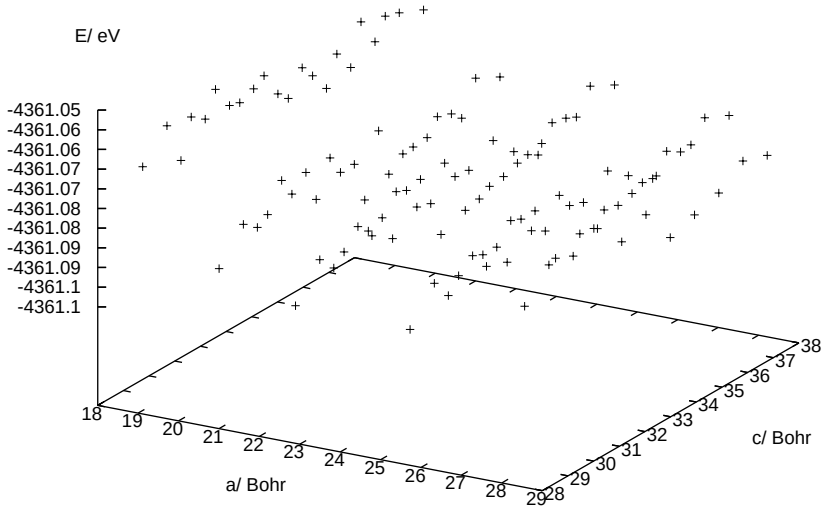


Figure 3 Energy v. box size (a=c) for calculations with a 100 Ry cutoff, showing periodic fluctuations in the energy with increase in box size, rather than the steady convergence expected

detailed sampling of the box dimensions that we were able to observe an unforeseen issue: it was found that the fineness of the grid over which the system's energy is calculated was insufficient for the PCB molecule, resulting in the periodic fluctuation in the energy corresponding to the distance between grid points that is shown in Figure 3. This is known as the 'eggbox' effect, and occurs when an inadequate grid fineness is used in calculations. While the mesh cutoff is always tested for convergence before starting production calculations, it was not expected that the eggbox effect would be observed so strongly in calculations of PCBs. The grid size was fine-tuned by running a number of suites of the box dimension parameter sweep calculations with different mesh cutoffs and searching for convergence.

For each of the box convergence runs it was only necessary to perform single point calculations on the system to determine the degree of interaction across the periodic boundary conditions, resulting in the large number of short calculations for which high-throughput *e*Science is particularly useful.

2.4 Calculating Torsional Energy

The structure of ortho-2,2'-dichloro biphenyl (Figure 2) was described in z-matrix format, so that the torsion angle could be constrained and, therefore, varied over 360˚. In the first instance it was considered adequate to sample every 5˚ over the rotation around the central carbon bond. These calculations were autogenerated and autosubmitted as previously described.

The only constraint on the system was the fixing of the torsion angle, so the positions of the other atoms were allowed to optimise around this fixed angle.

2.5 *e*Science Tools Used

The SIESTA input files for each suite of calculations were automatically generated using parameter sweep scripts which have been developed within the *e*Minerals project precisely to leverage the enhanced computing power made available through grid technology. These scripts are described in detail in [7], along with a detailed description of my_condor_submit (MCS), the submission script used in this work. However, a short description will be given here for clarity regarding this work.

The parameter sweep scripts use, as input, a template SIESTA input file, which can be modified to create all of the required input files, and a very simple configuration file. The configuration file is used to specify the input parameters that should be varied within the input file and the range and number of values over which they are to be varied. From this simple description of the problem space, running one command results in the creation of both a local and a logical Storage Resource Broker (SRB) directory structure, and the creation of the required SIESTA input files within this structure. Also, relevant MCS input files are created in the local directory structure.

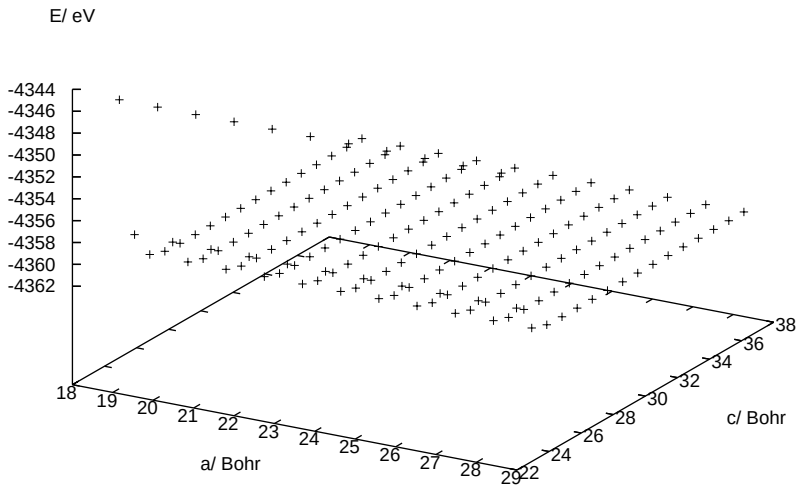


Figure 4 Energy v. box dimensions from calculations with a 300Ry cutoff and fcc + box centre grid sampling

The MCS files specify each of the jobs to be run, including the SRB location of the executable and input files, from where they should be downloaded and to where output files should be uploaded. In addition, the MCS files detail any relevant metadata to be collected. In the case of the box size convergence, the additional metadata requested were the lattice parameters, which were extracted from the XML SIESTA output file on completion of the calculation.

Once the directory structure has been created, jobs are submitted by running a second command which walks the local directory structure and submits all of the input files using MCS which takes care of all appropriate data management and the meta-scheduling over the available computing resources. This use of MCS as the underlying submission system means that we can maximise the use of available resources and minimise the latency between job submission and results retrieval.

The calculations are metascheduled across the *e*Minerals minigrid [8] using Globus to check the machine availability, and Condor-G to submit the jobs, around which MCS is wrapped to enable communications with the SRB and metadata collection / storage . The user has the option of specifying the machine on which the calculation is run, or whether to run on a condor pool or a cluster.

Such a large number of calculations as encountered here quickly becomes unmanageable and it becomes difficult to trace individual jobs or suites of calculations on the SRB file structure. Consequently, the use of metadata, and its automatic extraction from XML files using AgentX [9], a library for logically based xml data handling, is of paramount importance in such a combinatorial study as this one.

In general terms metadata is stored in three tiers: the study, which is a self-contained piece of work; the dataset; and data objects. Each study can be labelled with various topics from a controlled taxonomy and can be annotated with a high level description of this work. The study level metadata can be searched at a later date either via keywords within these annotations or via the topic labels. Searches can be performed using the RCommands, which can be run from the command line or the Metadata Manager, a web interface to the metatdata database. There tools are fully discussed in [10], but a description of the specifics of their use in this study follows.

In this case, prior to commencing the runs, a study for PCB molecules was created on the metadata database, and within this study each suite of calculations constituted one dataset. Each data object within the dataset relates to one directory containing the files for one calculation. The data object is associated with the URI for the directory within the SRB. As with the study, it is possible to annotate the dataset and data objects with descriptions, which can later be searched for keywords. In addition to these free text annotations, dataset and data objects can have parameters associated with them. These are arbitrary name value pairs, which can be used to index the data using key parameters of interest. If these parameters are numerical, it is possible to search for data objects or datasets that have a specified value for a certain parameter.

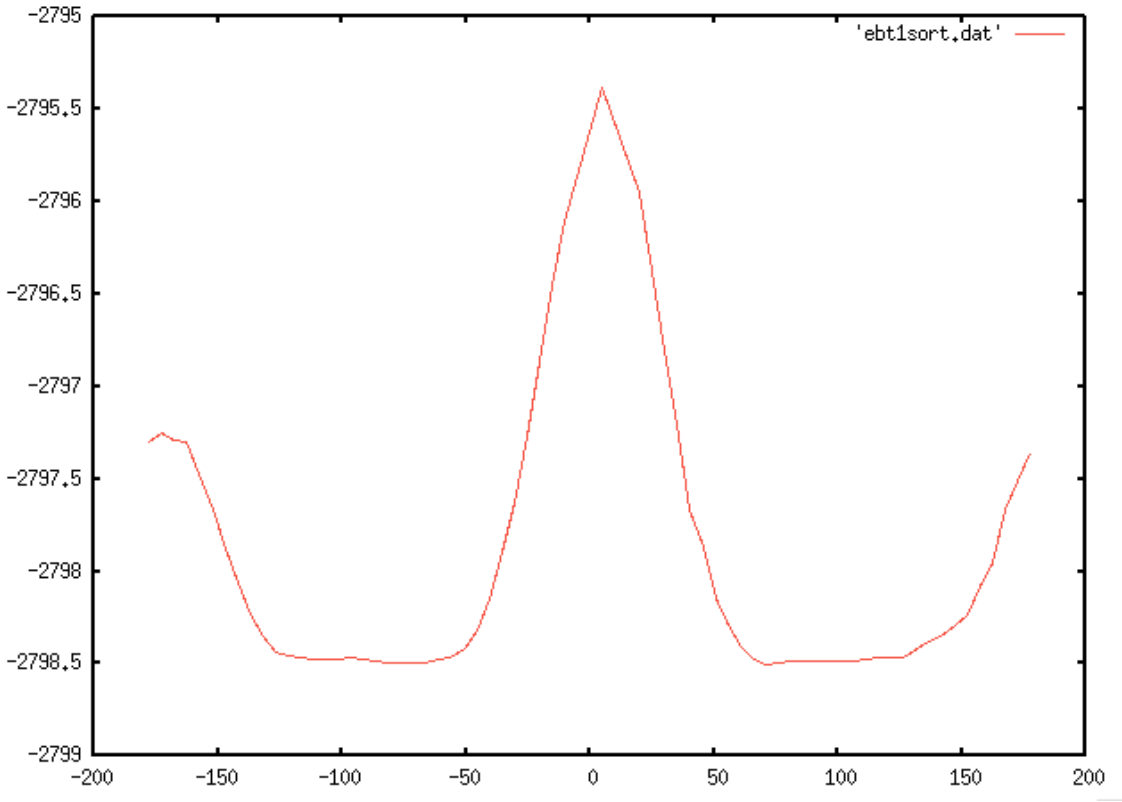


Figure 5 Energy (eV) v. torsion angle (degrees) with autogenerated basis set

In this work, each data object is associated with a number of parameters that are harvested on job completion by MCS, from both the MCS environment (including the name of the computer from which the jobs were submitted and the submission directory and the remote machine upon which the job has been run) and any metadata available from the simulation code (e.g.: the executable name and version). In addition, AgentX was used to harvest from the XML output file any metadata requested for extraction as specified in the configuration file used to generate the individual MCS scripts.

For the box convergence study, numerical values of the lattice vectors were stored as metadata, allowing the metadata to be searched for this parameter and a specified value. So, on completion of each calculation, the results are uploaded to the SRB and metadata is automatically updated. This removes the requirement from more traditional methods of logging into each remote machine to retrieve results, and making a record of each calculation in order to track the workflow. Obviously, with over one thousand jobs run in this study, such a method of working would be unmanageable, and prohibitive of this detailed type of parameterisation study.

With such a large number of files and suites as required for this study, and with the frequent updates of the simulation code used, it is of paramount importance to be able to trace the details of each calculation, both in order to process the results and to enable traceability of the workflow. This is important both for the individual scientist carrying out the calculations and for the facilitation of collaborations, which are integral to a virtual organisation as distributed as the *e*Minerals project.

The torsion angle calculations were constrained geometry optimisations. These calculations were generally well behaved; however, it was necessary to check the convergence of the calculations to ensure that the energy extracted from the XML output file would be the correct energy for the system. A number of python scripts and shell commands were used to check for convergence. Thereafter, an XSLT file was used to parse each XML file and extract the lattice vectors and total energy of the system, the data from which could then be plotted, and the images uploaded to the SRB for viewing by collaborators.

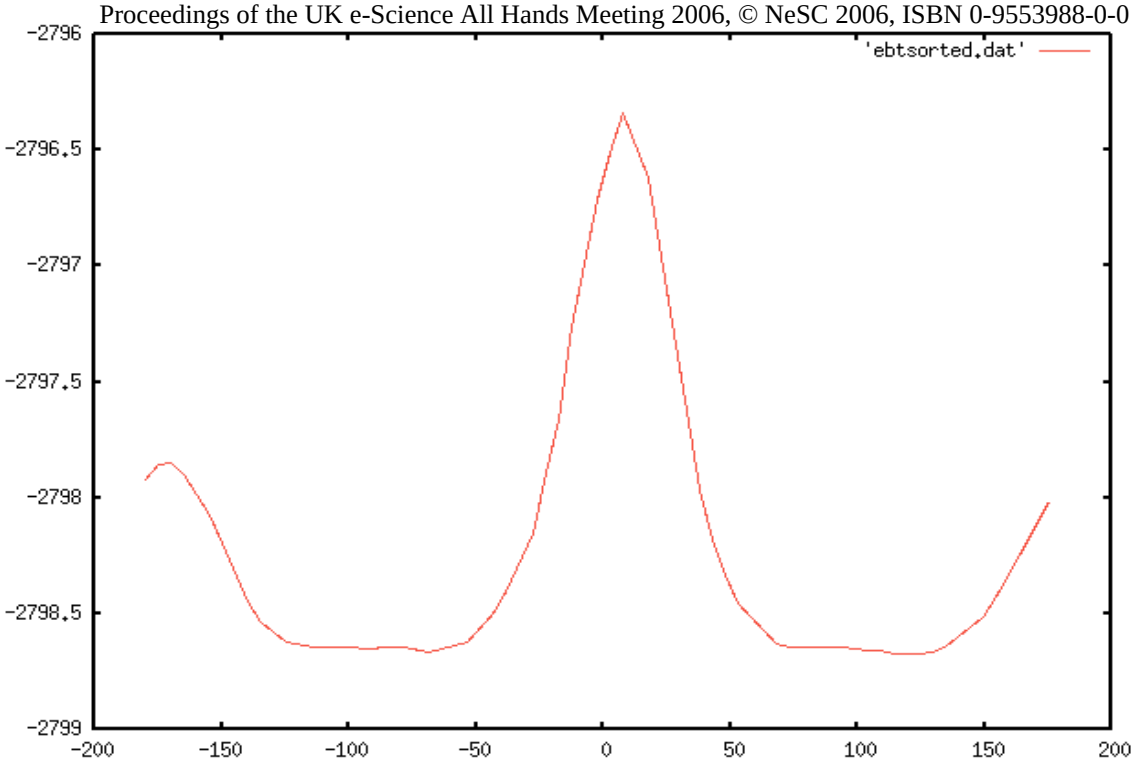


Figure 6 Energy (eV) v. torsion angle (degrees) with user-specified basis set

3. Results

3.1 Box Size Calculations

The first dimension sweep performed on the box containing the 6-PCB molecule showed wave-like perturbations in the energy with increasing box size (Figure 3). Further analysis revealed that energy minima coincide with grid points, such that the perturbations are synchronised with grid periodicity. Consequently, further suites of calculations were performed, 15 in total, in order to determine the least computationally expensive method to minimise this 'eggbox' effect. In addition to increasing the fineness of the grid, the grid sampling method was used in the hope of reducing the mesh cutoff needed to minimise these fluctuations. The graph showing the optimal combination of these factors, plotting box size against energy, is shown in Figure 4. The parameters decided upon were a 300 Ry cutoff and a combination of fcc and box centre grid cell sampling. It can be seen from the plot that a box size of 24 x 30 Bohr (roughly 13 x 16 Å) proved sufficient for removing size effects. The total number of calculations carried out to parameterise the model to such a high level of accuracy was 1815. Using conventional methods, this level of exploration of the parameter space would be prohibited by the hours required to generate and submit the calculations, and to process the results. It is probable that the 'eggbox' effect, so easily identified with the three-dimensional plots obtained *via* these calculations, would have gone unobserved, were traditional parameterisation methods used in the study. Usually, for such a study, the box size would be increased in fairly large increments, until it appears that the energy is near convergence. Thereafter, a few calculations that vary the lattice parameters with smaller increments would be carried out to tune the values, and convergence would be considered to have been achieved. It is conceivable that, using this method, the investigator could happen upon a well in the eggbox, and believe the parameters to be converged without seeing the larger picture and hence without realising that the mesh cutoff was insufficient for this study.

3.2 Torsional Energy

The energy of each calculation is plotted against the torsion angle in Figure 5. The calculations were carried out with an autogenerated DZP basis set as the first pass. The box size and mesh cutoff obtained from the box size parameterisation were used in the study to ensure that the calculations were of the highest accuracy in this respect.

The effect of different basis sets was to be determined in this study, in order to find the least computationally expensive, adequate description of the torsional rotation around the central C–C bond. As such, the autogeneration of the suites of calculations was extremely useful, as a change in the

basis set in the SIESTA input file was all that was required in order to generate and submit all the required jobs. At present, two basis sets have been tested. The autogenerated SIESTA DZP basis set, and a set of basis sets taken from previous parameterisation calculations for the PCDDs[1].

Inspection of Figure 5 shows that the barrier to rotation is highest at 0º, and that there is a minimum in the energy at around 70º, as expected, although this is only slightly lower in energy than the broad minimum within which it sits, between 70º and 130º. The kinetic barrier to rotation is around 1.25 eV at 180º and 3.25 at 0º, although a more detailed sampling is needed at 0º in order to determine this value accurately.

An analogous plot is shown for the explicit basis sets in Figure 6. It can be seen that, once again, the barrier to rotation is at 0º, and another at 180º. Once again an energy minimum is found to be at 70º, but a second minimum is apparent at roughly 120º. The barrier to rotation in this case, however, can be seen to be considerably lower at 2.5 eV. Further investigation is obviously necessary to ascertain the best description of the molecule, which will include the repetition of the suites of torsion angle calculations using different basis sets. Additionally, the *e*Science tools that have already been used can further be applied with different codes that are not limited to DFT calculations. Investigations into the applicability of DFT to the problem will be made by the study of the system using higher-level quantum mechanical calculations.

4. Conclusions

Quantum mechanical calculations have been carried out using *e*Science tools developed during the *e*Minerals project. These tools change the way that computational chemists are able to carry out scientific research for a number of reasons. First, the tools allow for facile generation of many files, removing the necessity for lengthy set-up times. Secondly, the metascheduling aspect of the setup and the use of the SRB for data storage, along with the use of digital certificates for security, removes the need for monitoring of the machines upon which the calculations are running, or the need to directly log into the machines. Searchable metadata, through the use of the RCommands and Metadata Manager, ensures that the vast numbers of calculations do not get confused and thereby remain manageable. The use of tools that test convergence in the jobs, along with the structure of XML output, which allows for the easy extraction of relevant data, means that the processing of the large number of results does not pose a problem to the research scientist.

The consequence of using all of these tools is that it is much easier to properly parameterise the models used, as a comprehensive sweep of parameter space is neither lengthy nor computationally too expensive when using high-throughput computing. In this particular case, the parameterisation led to the identification of a phenomenon that was not expected to be observed in this system, which might have gone unobserved using traditional methods.

5. Acknowledgments

We are grateful for funding from NERC (grant reference numbers NER/T/S/2001/00855, NE/C515698/1 and NE/C515704/1).

References:

1. TOH White, RP Bruin, J Wakelin, C Chapman, D Osborn, P Murray-Rust, E Artacho, MT Dove, M Calleja, *e*Science methods for the combinatorial chemistry problem of adsorption of pollutant organic molecules on mineral surfaces. *Proceedings of the All Hands Meeting, Nottingham, 773-780* (2005).
2. C Romming, HM Seip, and Aaneseno.Im, Structure of Gaseous and Crystalline 2,2'-Dichlorobiphenyl. *Acta Chemica Scandinavica Series A-Physical and Inorganic Chemistry A 28 (5), 507-514* (1974).
3. R Zimmermann, C Weickhardt, U Boesl, EW Schlag, Influence of chlorine substituent positions on the molecular structure and the torsional potentials of dichlorinated bipheyls: R2P1 spectra of the first singlet transition and AM1 calculations. *Journal of Molecular Structure 327, 81-997* (1994).
4. JM Soler, E Artacho, JD Gale, A Garcia, J Junquera, P Ordejon, D Anachez-Portal, *T*he SIESTA method for ab initio order-N materials simulation. *Journal Of Physics-Condensed Matter 14 (11), 2745-2779* (2002).
5. JP Perdew, K Burke, and M Ernzerhof, Generalized Gradient Approximation Made Simple. *Physical Review Letters 77, 3865-3868* (1996).
6. FH Allen, The Cambridge Structural Database: a quarter of a million crystal structures and rising. *Acta Crystallographica Section B-Structural Science 58, 380-388* (2002).
7. RP Bruin, TOH White, AM Walker, KF Austen, MT Dove, et al., Job submission to grid computing environments. *Submitted to All Hands Meeting 2006* (2006).

8. M Calleja, R Bruin, MG Tucker, MT Dove, R Tyer, L Blanshard, K Kleese Van Dam, RJ Allan, C Chapman, W Emmerich, P Wilson, J Brodholt, A Thandavan and VN Alexandrov, Collaborative grid infrastructure for molecular simulations: The *e*Minerals minigrid as a prototype integrated compute and data grid. *Molecular Simulation, 31, 5, 303-313 (2005)*
9. PA Couch, P Sherwood, S Sufi, IT Todorov, RJ Allan, PJ Knowles, RP Bruin, MT Dove and P Murray-Rust, Towards Data Integration for Computational Chemistry. *Proceedings of the All Hands Meeting*, Nottingham (2005).
10. RP Tyer, PA Couch, K Kleese van Dam, IT Todorov, RP Bruin, TOH White, AM Walker, KF Austen, MT Dove, MO Blanchard, Automatic metadata capture and grid computing. *Submitted to All Hands Meeting 2006* (2006)

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Anatomy of a grid-enabled molecular simulation study: the compressibility of amorphous silica

Andrew M Walker[1], Martin T Dove[1,2], Lucy A Sullivan[1], Kostya Trachenko[1], Richard P Bruin[1], Toby OH White[1], Peter Murray-Rust[3], Rik P Tyer[4], Phillip A Couch[4], Ilian T Todorov[1,4], William Smith[4], Kerstin Kleese van Dam[4]

1. *Department of Earth Sciences, University of Cambridge, Downing Street, Cambridge CB2 3EQ*
2. *National Institute for Environmental eScience, University of Cambridge, Downing Street, Cambridge CB2 3EQ*
3. *Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW*
4. *CCLRC, Daresbury Laboratory, Warrington, Cheshire WA4 4AD*

Abstract

We report a case study in grid computing with associated data and metadata management in which we have used molecular dynamics to investigate the anomalous compressibility maximum in amorphous silica. The primary advantage of grid computing is that it enables such an investigation to be performed as a highly-detailed sweep through the relevant parameter (pressure in this case); this is advantageous when looking for derived quantities that show unusual behaviour. However, this brings with it certain data management challenges. In this paper we discuss how we have used grid computing with data and metadata management tools to obtain new insights into the behaviour of amorphous silica under pressure.

Introduction

It is now well-established that grid computing comes into its own in the physical sciences when it enables simulation studies to be carried out across a sweep of the input parameters. Examples might be studies of a system as a function of external conditions such as temperature or pressure. Whilst the existence of a grid of computers facilitates the parallel running of many separate simulations, to make effective use of the potential of grid computing it is essential to have appropriate workflow and data management tools. In this paper we report on a case study that has used a set of tools developed within the *e*Minerals project.

The particular case concerns a study of the properties of amorphous silica (SiO_2) as a function of pressure. Our interest concerns the way that volume varies with pressure. In almost all materials, relative volume changes become smaller with increasing pressure, which is equivalent to the statement that most materials become stiffer under pressure. Usually this can be explained by the fact that the atoms are being squeezed closer together, and the closer they are the stiffer the structure. However, amorphous silica behaves differently. On increasing pressure, amorphous silica initially becomes softer, until it crosses over to normal behaviour [1]. Formally the stiffness is defined by the inverse of the compressibility, κ^{-1}, where

$$\kappa = -V^{-1}(\partial V/\partial P).$$

Here V is the volume, and P is the pressure. In most materials, κ decreases on increasing pressure, but in amorphous silica κ has a maximum at a pressure of around 2 GPa.

Our approach is to use the classical molecular dynamics simulation method to study the pressure-dependence of amorphous silica. Because we need to calculate a differential, it is important to obtain a large number of data points on the volume/pressure graph, and it is in this regard that grid computing plays an important role. Using a grid enables the many separate jobs to be run at the same time, increasing the throughput by more than an order of magnitude so that collecting many data points becomes a viable process.

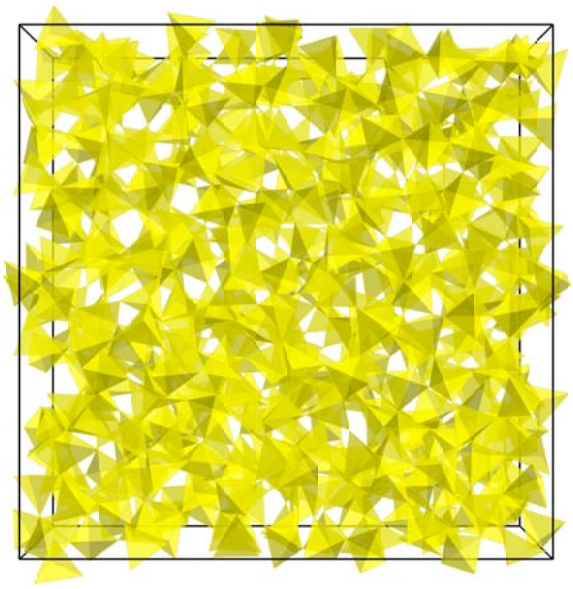


Figure 1. Configuration used in the simulations described in this paper, with SiO_4 polyhedra represented as tetrahedra rather than representing the individual atoms.

In this work we make use of the following technologies:

- Methods to create and submit many jobs in which one or more parameters are varied [2];
- Metascheduling within a minigrid compute environment, with jobs distributed over clusters and Condor pools [2,3];
- Use of the San Diego Storage Resource Broker (SRB) for data archiving and the sharing of data files [4];
- Use of XML output data and associated tools to aid data analysis and sharing of the information between collaborators [5];
- Incorporation of workflows within the job submission procedure to enable analysis to be performed on the fly [2];
- Automatic metadata capture using the recently-developed RCommands [6].

The purpose of this paper is to describe how these tools were combined to facilitate a detailed molecular dynamics simulation study of the compressibility of amorphous silica using grid computing.

Science background

Amorphous silica, SiO_2, is a random network of corner-linked SiO_4 tetrahedra, Figure 1. We work with configurations of 512 tetrahedra generated from initial configurations of amorphous elemental silicon [7] and tested against neutron total scattering data [8].

The issue of compressibility concerns the inherent flexibility of the network of connected tetrahedra. This is a subtle issue, because standard engineering methods of counting constraints and degrees of freedom do not capture the whole story. We have previously demonstrated [7] that the silica network has an inherent network flexibility in which the SiO_4 tetrahedra can rotate and buckle the network without the tetrahedra themselves needing to distort. Such motions will cost relatively little energy; the higher-energy processes are those that cause the SiO_4 tetrahedra to distort, either through bending of the O–Si–O bond angles or stretching of the Si–O bonds. There are two ways in which buckling of the network of corner-linked SiO_4 tetrahedra can happen. One is through fast vibrations, and the other is through larger jump motions in which several tetrahedra change their orientations together. Animations of both processes are available from references 9 and 10 respectively, and are surprisingly instructive.

Our approach is to consider the behaviour of amorphous silica in the two extremes of large negative and positive pressures in comparison with intermediate pressures. First we note that the compressibility, as defined earlier, can also be defined in terms of the second derivative of the free energy G:

$$\kappa = -V^{-1}(\partial^2 G/\partial P^2).$$

Thus compressibility is related to changes in energy, and our hypothetical extreme end states are both states in which any changes are necessarily accompanied by large changes in energy as compared to the intermediate state. At large negative pressures (corresponding to stretching the material) the bonds are themselves stretched tight and the flexibility of the network is accordingly reduced. To change the pressure in this extreme will involve distorting the SiO_4 tetrahedra – either by changing bond lengths or bond angles – which as noted above is quite a high energy process. At the high-pressure extreme, atoms are pushed tightly together and further changes in volume can again only be accomplished by distorting the SiO_4 tetrahedra. But in the intermediate region, where there is more flexibility of the network, volume changes can be accomplished by crumpling the network without any distortions of the SiO_4 tetrahedra. Since this is a low energy process, the compressibility is a lot higher.

The task we set ourselves was to demonstrate the reasonableness of this hypothesis, and the chosen tool is molecular dynamics simulation. We have two good sets of interatomic potential energy functions for silica, both based on quantum mechanical calculations of small clusters of silicon and oxygen atoms; these are described in references 11 and 12 respectively. For the present paper we will only present results using the model of reference 11, but will refer to the other set of results later. We use the DL_POLY_3 molecular dynamics simulation code [13], which has been adapted for working within a grid computing environment.

We ran simulations for pressures between ±5 GPa and at a temperature of 50 K. Our aim was to capture data for many pressures within this range, and to analyse the resultant configurations at the same time as running the simulations. At each pressure we ran one simulation with a constant-pressure and constant-temperature (*NPT*; *N* implies a constant number of atoms) algorithm in order to obtain data for the equilibrium volume, followed by a simulation using a constant-volume and constant-energy (*NVE*) algorithm for the more detailed analysis (see later in this paper). The analysis was performed using additional programs, and was incorporated within the workflow of each job carried out within the grid computing environment.

eScience methodology

Grid computing environment

The simulations were performed on the combination of the *e*Minerals minigrid [3,14], which primarily consists of linux clusters running PBS, and CamGrid [15], which consists of flocked Condor pools. Although the *e*Minerals minigrid and CamGrid are independent grid environments, they have overlapping resources, and the tools developed to access the *e*Minerals minigrid [3,14] have been adapted to work on CamGrid, and, incidentally, to also enable access to NGS resources, in the same manner.

Access to the *e*Minerals minigrid is controlled by the use of escience digital certificates and the use of the Globus toolkit. We have developed a metascheduling job submission tool called `my_condor_submit` (MCS) to make the process easier for end users [2,3,15]. This uses the Condor-G interface to Globus to submit three separate jobs per simulation. The first job takes care of data staging from the SRB to the remote computing resource, whilst the last job uploads output data back to the SRB as well as capturing and storing metadata. The middle job takes care of running the actual simulation and corresponding analysis as part of a script job.

Data management

Data management within the *e*Minerals minigrid is focussed on the use of the San Diego Storage Resource Broker. The SRB provides a good solution to the problem of getting data into and out of the *e*Minerals minigrid. However, it also facilitates archiving a complete set of files associated with any simulation run on the minigrid. The way that the *e*Minerals project uses the SRB follows a simple workflow:

1. The data for a run, and the simulation executable, are placed within the SRB. It is not necessary for the files to be within the same SRB collection.
2. A submitted job, when it reaches the execute machine, first downloads the relevant files.
3. The job runs, generating new data files.
4. At the end of the job, the data files generated by the run are parsed for key metadata which is ingested into the central metadata datbase, and the files are put into the SRB for inspection at a later time.

This simple workflow is managed by the MCS tool using Condor's DAGman functionality. We have built into the workflow a degree of fault tolerance, repeating any tasks that fail due to unforseen errors such as network interruption.

Combinatorial job preparation tools

One of the key tasks for escience is to provide the tools to enable scientists to access the potential benefits of grid computing. If scientists are to run large combinatorial studies routinely, they need tools to make setting-up, submitting, managing and tracking of many jobs nearly as easy as running a single job. To enable the scientists within the *e*Minerals project team to run detailed combinatorial studies one of the authors (RPB) has developed a set of tools that generate and populate new data collections on the SRB for all points on the parameter sweep, and then generate and submit the set of MCS scripts [2].

All that is required from the user is to provide a template input file from which all

necessary unique input files will be created. The user specifies the name of the parameter whose value is to be varied, the start and end values, and the number of increments. For example, a user could specify that they wish to vary pressure between –5 and +5 GPa in 11 steps which would result in input files being created for pressures of –5, –4, –3, ... +4, +5 GPa. The tools then create a simple directory structure, with each sub-directory containing the necessary input files. A similar directory structure will also be created on the SRB. Each directory on the submission machine also contains the relevant MCS input script, with appropriate SRB calls and metadata commands.

The subsequent stage is job submission. The user runs one command, which walks through the locally created directory structure and submits all of the jobs using MCS. It is MCS that takes care of the issue of deciding where to run the simulation within the *e*Minerals minigrid or Camgrid environments, using a metascheduling algorithm explained elsewhere [2]. This algorithm ensures that the user's simulations do not sit in machine queues for longer than is necessary, and that the task can take full advantage of all available resources.

The tools also track the success of the job submission process, and any errors that occur as part of the submission are recognised. For example, if one of the Globus gatekeepers stops responding, it will cause Condor-G to fail in the submission. This will be noticed, and a user command will provide information on all failed jobs and will resubmit them as appropriate.

Workflow within the job script

Usually MCS is used to submit a binary executable, but can also submit a script (eg shell or Perl) containing a set of program calls. In our application, a standards-compliant shell script is used to control the execution of a number of statically linked executables and a simple Perl script to run the analysis. In detail the script runs the following codes in order on the remote host: first DL_POLY_3 is run with the *NPT* ensemble in order to generate a model with the appropriate density for the pressure of interest, then the output of this run is used as input for a second DL_POLY_3 run in the *NVE* ensemble in order to sample the vibrational behaviour of the system at this density. Following the second molecular dynamics simulation, pair distribution functions are extracted for the Si–Si, Si–O and O–O separations, and configurations are extracted for analysis of the atomic motions based on a comparison of all configurations. Finally a Perl script collates the results of this analysis for later plotting. This analysis will not be reported in this paper, but is mentioned here to make the extent of the workflow clear.

Use of XML

The *e*Minerals project has made a lot of use of the Chemical Markup Language (CML) to represent simulation output data [5]. We have written a number of libraries for using CML within Fortran codes (most of our simulation codes are written in Fortran), and most of our simulation codes now write CML. We typically write three blocks of CML, one for "standard" metadata (some Dublin Core items, some specific to the code version and compilation etc), one to mirror input parameters (such as the input temperature and pressure, technical parameters such as cut-off limits), and one for output properties (including all computed properties step by step and averages over all steps). This is illustrated in Figure 2.

The XML files stored within the SRB are transformed to HTML files using the TobysSRB web interface to the SRB, with embedded SVG plots of the step-by-step output [16]. This enables the user to quickly inspect the output from runs to check issues such as convergence.

XML output also allows us to easily collect the desired metadata related to both input and output parameters using the AgentX library, developed by one of the authors (PAC) [17]. This has been integrated into the MCS workflow, enabling the user to easily specify the metadata they wish to collect as part of their job submission process without needing to know the ins and outs of XML file formats.

Results management

When many jobs are run as part of a single study, it is essential to have tools that collate the key results from each run. For example, one key output from our work will be a plot of volume against pressure, with each data point obtained from a single computation performed using grid computing. In this we exploit the use of XML in our output files, because the required averaged quantities can be accessed by retrieving the value of the relevant XML element as per Figure 2. This value can be retrieved for each of the individual files using a simple XSLT transform, combining all of the values together

```
<?xml version="1.0" encoding="UTF-8"?>
<cml xmlns="http://www.xml-cml.org/schema"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:dc="http://purl.org/dc/elements/1.1/title"
 xmlns:dl_poly:"http://www.cse.clrc.ac.uk/msi/software/DL_POLY/dict"
 xmlns:dl_polyUnits:"http://www.cse.clrc.ac.uk/msi/software/DL_POLY/units">

<metadataList>
 <metadata name="dc:contributor" content="I.T.Todorov &amp;amp; W.Smith"/>
 <metadata name="dc:source"
 content="cclrc/ccp5 program library package, daresbury laboratory molecular dynamics
program for large systems"/>
 <metadata name="identifier" content="DL_POLY version 3.06 / March 2006"/>
 <metadata name="systemName" content="DL_POLY : Glass 512 tetrahedra"/>
</metadataList>

<parameterList title="control parameters">
 <parameter title="simulation temperature" name="simulation temperature"
 dictRef="dl_poly:temperature">
 <scalar dataType="xsd:double" units="dl_polyUnits:K"> 5.0000E+01 </scalar>
 </parameter>
 <parameter title="simulation pressure" name="simulation pressure"
 dictRef="dl_poly:pressure">
 <scalar dataType="xsd:double" units="dl_polyUnits:katms"> -3.0000E+01 </scalar>
 </parameter>
 <parameter title="simulation length" name="selected number of timesteps"
 dictRef="dl_poly:steps">
 <scalar dataType="xsd:integer" units="dl_polyUnits:steps"> 50000 </scalar>
 </parameter>
</parameterList>

 <propertyList title="rolling averages">
  <property title="total energy" dictRef="dl_poly:eng_tot">
   <scalar dataType="xsd:double" units="dl_polyUnits:eV_mol.-1"> -2.7360E+04 </scalar>
  </property>
  <property title="volume" dictRef="dl_poly:volume">
   <scalar units="dl_polyUnits:Angstroms.3">2.2316E+04</scalar>
  </property>
 </propertyList>
 <propertyList title="execution time">
  <property title="run time">
   <scalar dataType="xsd:double" units="dl_polyUnits:s"> 17475.422 </scalar>
  </property>
 </propertyList>
</cml>
```

Figure 2. Example CML output from DL_POLY showing the key data lists.

then results in a list of points. This can easily be plotted as a graph using a further XSLT transformation into an SVG file.

These transformations can be done very quickly, and more importantly they can be done automatically, which means that the user and his/her collaborators simply need to look at a graph in a web page to quickly analyse trends within the data, rather than having to open hundreds of files by hand to find the relevant data values to then copy into a graph plotting package. In our experience, this is the sort of thing that makes grid computing on this scale actually usable for the end user, and facilitates collaborations.

Metadata

With such quantities of data, it is essential that the data files are saved with appropriate metadata to enable files to be understood and data to be located using search tools. We have developed tools, called the RCommands, which

MCS use to automatically collect and store metadata [6]. The metadata are culled from the CML output files, collected from the metadata, parameter and property lists. Property data such as the computed average volume are used as metadata because they will be parameters within the metadata against which users are able to run search commands on.

Other information, including metadata content regarding the execution machine and directory, as well as the default metadata created as part of any of our XML output file creation, are also captured. These items of metadata provide a valuable audit trail for each simulation, in part replacing the scientist's traditional log book with a much more searchable and efficient means of tracking their submitted simulations. Scientists cannot reasonably be expected to keep track of each of several hundred simulations performed in this way without efficient and automatic metadata capture, since doing so by hand would result in more time being spent recording this sort of information than in actually analysing the science identified by the results.

It should be noted that the use of the RCommands and the AgentX tools did not give a significant overhead to the running of the the jobs. The DL_POLY_3 simulations typically took 8 hours, whereas the metadata extraction only took 30 minutes or less.

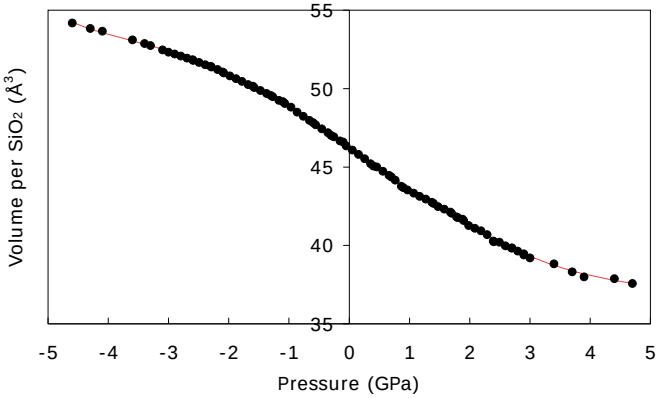


Figure 3. Pressure dependence of the volume (points) of amorphous silica, fitted with a polynomial (red curve).

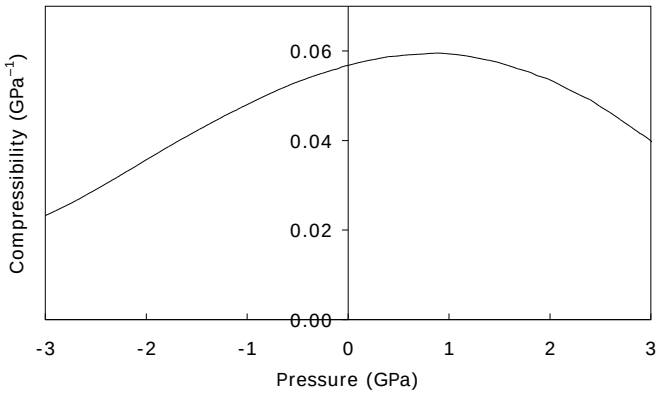


Figure 4. Compressibility of amorphous silica generated by differential of the fitted polynomial to the simulation volume.

Results

Volume curve

The first key result we were aiming at was a plot of volume *vs* pressure, which is show in Figure 3. What is clear from this diagram is that the slope of the graph is greatest for intermediate pressures, indicating that amorphous silica is softest around ambient pressures. The virtue of having many points is that we were able to fit a polynomial to the data in order to extract the derivative dV/dP with reasonable confidence. We plot the compressibility, $\kappa = -V^{-1}(\mathrm{d}V/\mathrm{d}P)$ in Figure 4. The maximum in the compressibility occurs at a pressure of around 1 GPa. This is a bit less than the experimental value (2 GPa), but given that this is a second-order effect, the difference between experiment and simulation is not significant. What is important from this plot is that we have successfully modelled the compressibility maximum in amorphous silica, and we note here that we have reproduced this result with the model interatomic potential of reference [12] as well. This implies that the compressibility maximum is not a subtle feature peculiar to the details of the interatomic potentials (real or model) but is a consequence of the nature of the structure and atomic connectivities.

Interatomic distances

The distribution of interatomic distances is described by the pair distribution function, $g(r)$, such that the number of pairs of atoms with separation between r and r + dr is given by $4\pi r^2 g(\mathrm{r})\mathrm{d}r$. $g(r)$ is thus a normalised histogram of interatomic distances, which for close neighbours is typically peaked around a well-defined mean separation. The mean separations for Si–O, O–O and Si–Si nearest neighbours are

plotted in Figure 5, normalised to unity at zero pressure to facilitate comparison. The Si–O and O–O distances are defined by the relatively-rigid SiO_4 tetrahedra – the ratio of the O–O to Si–O mean distance is equal to $(8/3)^{1/3}$ – and it can be seen that their mean distances barely change with pressure. Thus we see that the SiO_4 tetrahedra on average barely change their size or shape across the range of pressures. On the other hand, the mean Si–Si distance varies much more with pressure, almost scaling with the length scale of the simulation sample, i.e. the cube root of the sample volume (also shown in Figure 5). The pressure-dependence of the Si–Si distance suggests that the network is buckling with a folding of groups of connected tetrahedra.

The variances of the distributions of specific interatomic distances in $g(r)$ are shown in Figure 6. These show a number of interesting features. Most clear is that the Si–O distribution is very sharp (small variance); this reflects the fact that this is a very strong bond with a high frequency, and hence low amplitude, stretching vibration. The variance of the Si–Si distribution clearly increases on increasing pressure, consistent with bucking of the network. The variance of the O–O distribution is interesting. Although the mean distance (figure 5) varies no more than the mean Si–O distance, there is greater variation of the O–O distance. This means that all deformations of the SiO_4 tetrahedra mostly involve flexing of the O–Si–O bond angles. It is interesting to note that the variance of the O–O distances has a minimum around the pressure of the maximum in the compressibility.

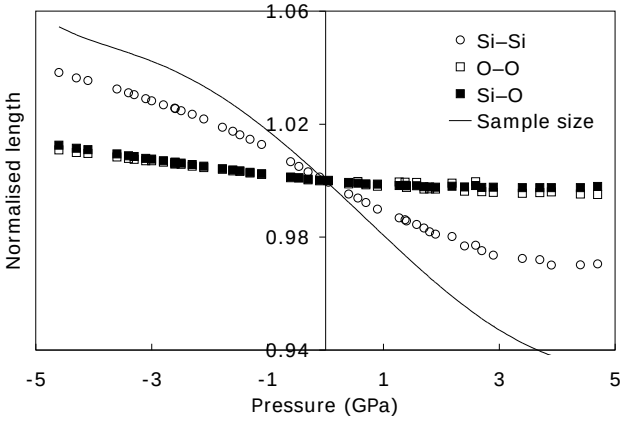


Figure 5. Comparison of the mean interatomic nearest-neighbour distances, normalised to unity at zero pressure in order to aid comparison of the dependence on pressure. The curve is the cube root of the volume, and represents the linear size of the simulation sample.

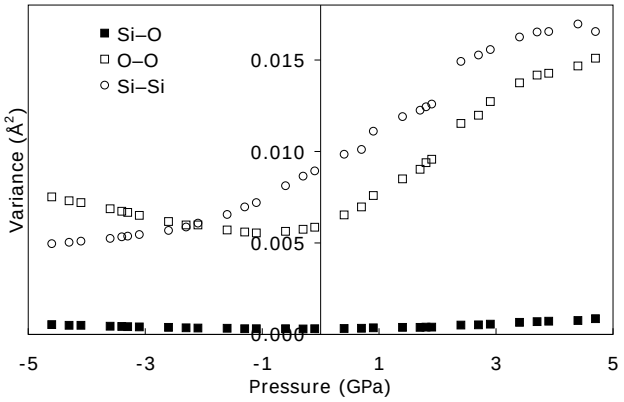


Figure 6. Pressure dependence of the variances of the distributions of nearest-neighbour interatomic distances.

Summary points

- We have seen how grid computing enables us to run many concurrent simulations, which enable us to obtain data with sufficient points to be able to extract derived quantities that show anomalous behaviour.
- Launching many jobs is a challenge that requires an automated solution. This work has been carried out using the parameter sweep tools developed within the *e*Minerals project.
- The actual mechanism of submitting the jobs to Globus resources and then managing the workflows, including interaction with the SRB, was enabled using the MCS tool developed by the *e*Minerals project.
- Data management presented several challenges. The use of the SRB for data storage was already in use within the *e*Minerals project, and this case study showed the value of the SRB for data archiving.
- This work was greatly helped by the use of XML (CML) file outputs. Examples were the use of the TobysSRB tool to inspect file output stored in the SRB in XML format, the use of XML to gather key data from many data files stored in the SRB, and the use of the SRB in gathering metadata.
- With the large number of data files generated in this study, automatic metadata capture is essential. We have used the RCommand framework as used within the MCS tool together with the Rparse tool to add metadata associated with each set of files.

- This work was collaborative in nature, involving at times sharing of data between colleagues who were in physically different locations. Sharing of data was enabled using the above tools and methods.
- Finally, it is worth remarking that a large part of the simulation work reported in this study was carried out by a third year undergraduate student (LAS) as her degree project. We were confident, as proved to be the case, that the escience tools developed to support this work could easily be picked up by someone with no previous exposure to escience or computational science and who was working under some considerable time pressure.

Acknowledgements

We are grateful for funding from NERC (grant reference numbers NER/T/S/2001/00855, NE/C515698/1 and NE/C515704/1).

References

1. OB Tsiok, VV Brazhkin, AG Lyapin, LG Khvostantsev. Logarithmic kinetics of the amorphous-amorphous transformations in SiO_2 and GeO_2 glasses under high-pressure. *Phys. Rev. Lett*. **80**, 999–1002 (1998)
2. RP Bruin, TOH White, AM Walker, KF Austen, MT Dove, RP Tyer, PA Couch, IT Todorov, MO Blanchard. Job submission to grid computing environments. *Proceedings of All Hands 2006*
3. M Calleja, R Bruin, MG Tucker, MT Dove, R Tyer, L Blanshard, K Kleese van Dam, RJ Allan, C Chapman, W Emmerich, P Wilson, J Brodholt, A.Thandavan, VN Alexandrov. Collaborative grid infrastructure for molecular simulations: The *e*Minerals minigrid as a prototype integrated compute and data grid. *Mol. Simul*. **31**, 303–313 (2005)
4. RW Moore and C Baru. Virtualization services for data grids. in *Grid Computing: Making The Global Infrastructure a Reality*, (ed Berman F, Hey AJG and Fox G, John Wiley), Chapter 16 (2003)
5. TOH White, P Murray-Rust, PA Couch, RP Tyer, RP Bruin, MT Dove, IT Todorov, SC Parker. Development and Use of CML in the *e*Minerals project. *Proceedings of All Hands 2006*
6. R.P.Tyer, P.A. Couch, K Kleese van Dam, IT Todorov, RP Bruin, TOH White, AM Walker, KF Austen, MT Dove, MO Blanchard. Automatic metadata capture and grid computing. *Proceedings of All Hands 2006*
7. KO Trachenko, MT Dove, MJ Harris, V Heine. Dynamics of silica glass: two-level tunnelling states and low-energy floppy modes. *J Phys.: Cond. Matt*. **12**, 8041– 8064 (2000)
8. MG Tucker, DA Keen, MT Dove, K Trachenko. Refinement of the Si–O–Si bond angle distribution in vitreous silica. *J Phys.: Cond. Matt*. **17**, S67–S75 (2005)
9. http://www.esc.cam.ac.uk/movies/trid4t.html
10. http://www.esc.cam.ac.uk/movies/glass1.html
11. S Tsuenyuki, M Tsukada, H Aoki, Y Matsui. 1st principles interatomic potential of silica applied to molecular dynamics. *Phys. Rev. Lett*. **61**, 869–872 (1988)
12. BWH van Beest, GJ Kramer, RW van Santen. Force fields for silicas and aluminophosphates based on *ab initio* calculations. *Phys. Rev. Lett*. **64**, 1955–1958 (1990)
13. IT Todorov and W Smith. DL_POLY_3: the CCP5 national UK code for molecular-dynamics simulations, *Roy. Soc. Phil. Trans*. **362**, 1835–1852 (2004)
14. M Calleja, L Blanshard, R Bruin, C Chapman, A Thandavan, R Tyer, P Wilson, V Alexandrov, RJ Allen, J Brodholt, MT Dove, W Emmerich, K Kleese van Dam. Grid tool integration within the eMinerals project. P*roceedings of the UK e-Science All Hands Meeting 2004*, (ISBN 1-904425-21-6), pp 812–817
15. M Calleja, B Beckles, M Keegan, MA Hayes, A Parker, MT Dove. CamGrid: Experiences in constructing a university-wide, Condor-based grid at the University of Cambridge. *Proceedings of the UK e-Science All Hands Meeting 2004*, (ISBN 1-904425-21-6), pp 173–178
16. TOH White, RP Tyer, RP Bruin, MT Dove, KF Austen. A lightweight, scriptable, web-based frontend to the SRB. *Proceedings of All Hands 2006*
17. PA Couch, P Sherwood, S Sufi, IT Todorov, RJ Allan, PJ Knowles, RP Bruin, MT Dove, P Murray-Rust. Towards data integration for computational chemistry. *Proceedings of All Hands 2005* (ISBN 1-904425-53-4), pp 426–432

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Analysis and Outcomes of the Grid-Enabled Engineering Body Scanner

Kevin T. W. Tan[1], **Daniela Tsaneva[2]**, Michael W. Daley[2], Nick J. Avis[2], Philip J. Withers[1]
[1]Manchester Materials Science, University of Manchester, Grosvenor Street, Manchester M1 7HS, UK
[1]Corresponding Fax: +44(0)-161-306-3586
[2]School of Computer Science, Cardiff University, Queen's Buildings, Newport Road,
P.O. Box 916, Cardiff CF24 3XF, UK

Abstract

This paper presents initial analysis and outcomes from the Integrated & Steering of Multi-Site Experiments for the Engineering Body Scanner (ISME) Virtual Research Environment (VRE) project to enable geographically remote teams of material scientists to work together. Comparisons between different AccessGrid (AG) tools have been made, allowing us to identify their advantages and disadvantages regarding quality (visual and audio) over broadband networks, utility, ease of use, reliability, cost and support. These comparisons are particularly important when considering our intended use of AG in non-traditional, non-office based settings, including using slow domestic broadband networks to run AG sessions from home to support experiments at unsociable hours. Our detailed analysis has informed the development of a suite of services on the web portal. Besides user interactions, a suite of services based on JSR-168 had been developed for evaluation. Each service has been evaluated by materials scientist users, thereby allowing qualitative user requirement gathering. This feedback has resulted in a prioritised implementation list for these services within the prototype web portal allowing the project to develop useful features for material scientists and the wider VRE community.

1. Background

The safe lifetime of a manufactured component can not normally be predicted simply by knowing its geometry and the stresses applied to it externally. Also important are any defects or flaws as well as residual stresses that self-equilibrate with the component. These residual stresses arise as a consequence of manufacture and previous service and add to any externally applied stresses. A number of complementary material characterisation techniques have been developed capable of providing 3D maps of structure and stress inside engineering components. This inside information allows the integrity and lifetime of the structure to be predicted. The fact that the experimental methods are non-invasive means that the evolution of structure in response to service conditions (static loads, fatigue, stress corrosion environments, etc) can be followed in real time through life.

Experiments to measure residual stresses often take place at International User Facilities (Figure 1), where the experimenters must work at speed, continuously during 24 hours/day experiments. Beam time is very precious and it may be months before another time slot is available to the team. Limitations associated with travel mean that often post-doctoral researchers and/or PhD students travel to the site and must work in shifts alongside instrument scientists. Furthermore, key decision points are often encountered late at night and without the benefit of a preliminary analysis of the collected data. Due to inexperience/tiredness simple mistakes are sometimes made which are only revealed subsequently off-site upon detailed analysis and too late to rectify the situation by modifying the set-up on-site.

Telephone calls and email have been the primary methods to support remote discussions between the experimental site and the scientist's university or home to explain the problems encountered and to utilise the expertise knowledge. The asynchronous nature of email can dislocate the exchange of ideas and renders brainstorming impossible. Experimenters also need to discuss the newly obtained experimental results, often by sharing customised analysis applications (2D and 3D visualisation) between experiment site and university / home. It is therefore, preferable to allow the various participants to log-in to a shared server which hosts the required applications to support common data analysis, visualization and discussions. Experimental thinking time is precious and so the interaction infrastructures must be natural and unobtrusive to gain wide-spread acceptance. Furthermore the infrastructure must be capable of supporting users in non-traditional non-office based situations – for instance the scientist at home connecting using domestic broadband services.

We have previously presented the need for advanced collaborative tools to support residual stress measurement experiments [1]. This paper describes the work in progress of the current JISC-VRE project, Integrated & Steering of Multi-site Experiments for the Engineering Body Scanner (ISME) focussing on the:

1). evaluation of AG tools to assist communication between remotely located experimenters and teams at university / home;

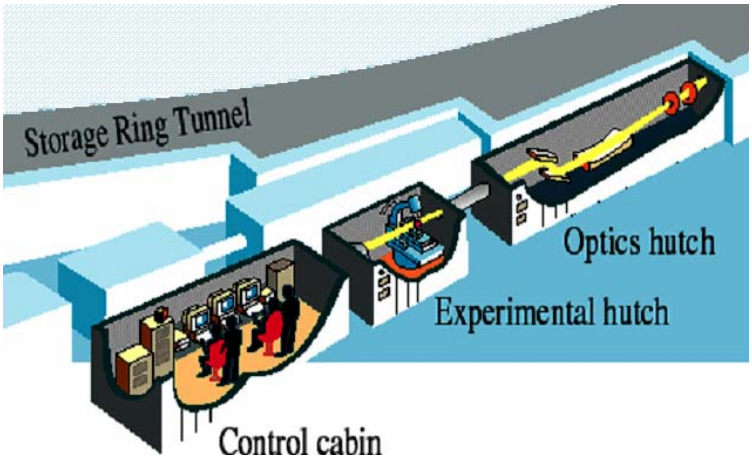


Figure 1: A beamline at an international Facility (ESRF)

2). analysis of various web services required in a web portal to aid and assist experiment/data management.

It is important to remember this project is primarily about examining the extent to which VREs can assist distributed teams carry out experiments, rather than to develop novel or sophisticated VRE tools. Our focus is very much on the seamless and transparent integration and takes up of low cost/accessible infrastructure technology and the identification of the most important tools and their role/impact in supporting 24 hour experiments at international facilities.

2. Existing ISME System

The ISME project is funded as part of the JISC-VRE Programme with the aim of integrating and refining existing tools into a VRE to make them deployable by teams of instrument scientists, material scientists and engineers in a transparent and robust manner. It is helping to extend the culture and functionality of collaborative multi-site experiments. Whilst this project is partly concerned with the technicalities of VRE-based development, the primary focus is very much on the end-user engineering community and the usability of the developed VRE. Consequently it is end user led and focuses on lightweight unobtrusive structures. Lessons learnt from applying these VRE tools to this specific focused group will have benefits for the wider science base using International User Facilities. The project targeted two sets of problems that require separate, but connected approaches.

- Human Interaction (Experimental Steering)
  - the need for a mechanism/medium for experiment steering, to discuss progress, modify strategies, and to train and instruct students
  - these aspects are being pursued via the provision of Access Grid (AG) functionality at the remote sites and universities / home
- Software Interaction (Data Management)
  - the need for a mechanism/medium for collaboratively analysing data and making archival data, collected elsewhere or during previous experiments, available for immediate side-by-side comparisons.
  - these activities involve embedding a set of well defined web services within a portal service framework using toolsets such as uPortal [3].

2.1 Experimental Steering Function

Stress measurement often takes place at remote sites and expert advice is often needed out-of-office hours. Unless the expert (e.g. PhD supervisor) accompanies the experimenters (e.g. PhD students) throughout, communications can be inadequate leading to avoidable errors and missed learning opportunities. Intelligent discussion, effective training and steering require a combination of three modality streams on screen:

- Group-based face to face contact, or at least voice to voice (via AG)
- A shared view of the experimental configuration (using AG)
- A common shared ‘tablet’ or ‘workspace’ to visualise results from the Data Management Function.

2.2 Data Management Function

Some logistical problems only become apparent when the experimenter tries to mount the sample on the instrument; these could be avoided by a virtual dry-run ahead of time. Once the experiment has begun the software required to assimilate the data may not be available at the work-cell or remote facility (Figure 1) usually because of computing, software or time constraints.

Discussions at this point, involving project supervisors and even industrial engineers based

on recently acquired and analysed real-time data could add true value to the experiments, especially if previous results can also be drawn down for comparison. In effect, all the collaborators will have opportunities to access and analyse the data and hence to offer their opinions on the particular measurement strategy. As a result the onus for experimentation and analysis can be more evenly shared amongst the group and a wider, multidisciplinary view brought to bear on the experiment in a timely fashion. Hence the learning curve for the PhD students can be much steeper.

We aim to extend our present ontology to support the more complex workflows and interactions that need to take place in our multidisciplinary teams. It is envisaged that this will lead to a cultural change in the way experiments are undertaken. Most importantly, it will allow the experimenter to regain the initiative. Moving away from pre-determined experimental sequences to interactive experiments, in which developments are monitored during the experiment and strategies modified accordingly. This project looks to tailor these toolkits for multi-site engineering experiments. The project is investigating the use of various common web portals to allow the VRE project to deploy the Engineering Body Scanner (EBS) [2] toolkit at multiple sites.

3. Analysis and Outcome of Experiment Steering Trials

The culture of the meeting room has been altered to accommodate the practical requirements of the end-users' for this project. The interactions via the multimedia resource (use of AG software) should not be at the meeting room level but at the experimental level, whereby the whole team can 'meet' together bringing their own data, modelling predictions and discuss and develop an evolving experimental strategy with Grid middleware. This task is not one of just teleconferencing, but rather a means of involving extended multi-disciplinary groups in the experimental steering process. It allows out-of-office hours steering using home broadband networks to link to the experimental site. We have analysed various AG software to identify the best options dependant on cost, ease of use and reliability regarding future use within the materials science community.

For the Experimental Steering Function we have trialled Access Grid, focusing primarily on how best to configure it to optimise HCI and usability. To this end we have established our own 'virtual venue'. Due to the 24 hour nature of our experiments and the cost and infrastructural implications it was deemed more appropriate to use Access Grid predominantly on computers with good quality webcams rather than via specialist traditional dedicated and fixed Access Grid studios. This is because firstly, for the experimenter, involvement must be seamless with the practical experimental function and secondly, because academics may need to enter into dialogue at home during unsociable hours. Connectivity between our two functionalities is achieved through the use of a shared virtual screen ("Shared Workspace") on which data analysed in the Data Function can be viewed on the web portal through the use of a standard web browser.

The initial trial of AG steering was between the Manchester School of Materials, Daresbury Laboratory and ISIS, Oxford. Common AG meeting tools include:

1. inSORS [4],
2. AccessGrid ToolKit 2.4 (AGTk) [5],
3. Videoconferecing Tools & Robus Audio Tools (vic&rat) [6]
4. Virtual Rooms VideoConferencing System (VRVS) [7]

Each has its own advantages and disadvantages in terms of the features provided, quality and cost. Although **vic&rat** are two separate video and audio applications, together they can considered as a basic AG tool. vic&rat have a command-line interface which requires users to know the multicast-unicast bridge server before use. The combination of command line interface and the need for a-priori knowledge severely and negatively impact on vic&rat as a general choice as an AG meeting tool.

VRVS, on the other hand, is a fully web and Applet-based AG tool using its own improved version of vic&rat incorporated with its own Java proxy applications. It allows almost anyone to join an AG meeting behind a firewall network without any ports having to be specially opened. Although VRVS can be in many respects be considered an ideal tool, limited communication between VRVS venues with existing AG virtual venues has relegated its utility for our purposes.

The choice between **inSORS** and **AGTk** can be informed by investigating the following attributes and comparisons discussed in Table 1:

a. **Audio & Video communication network**
   Focusing on support for AG meetings using a slow upload rate characteristic of home broadband networks.
b. **Ease of Use and Available Features**
   Focusing on the easiness and useful features provided by the different AG tools
c. **Reliability**
   Focusing on the stability of the "bridge server" supported by the AG tools.
d. **Support of Hardware Tools**
   Focusing on the wider range of hardware tools particularly to webcam and sound card.

e. Firewall Network
Support of network behind firewall routers typical of large international experimental facilities.

f. Cost & Set-Up & Support
Focus on the cost and after sales support for installation of AG tools.

Table 1: Comparisons between inSORS and AGTk

Aspects	inSORS	AGTk
(a)	Audio – supports 64Kb/s outgoing, ideal for broadband upload speed. – experiments indicated that it only supports Linear-16 encoding incoming audio from other AG tools Video – supports configurable video quality but a minimum 128Kb/s outgoing, not ideal for broadband upload speed.	Audio (based on vic&rat) – supports GSM encoding 24Kb/s outgoing but NOT supported by inSORS client. – supports Linear-16bit outgoing audio but it will take up to 256Kb/s audio, not ideal for broadband upload speed. Video – Configurable video quality with reasonable of 64Kb/s, ideal for broadband speed.
(b)	There are a suite of features supported within inSORS: – **IGMeeting** runs in the background, allowing other inSORS users to contact to set-up a meeting as required. – Meeting can be conducted by creating your own meeting venue without needing to use any of the pre-defined virtual venues. – **IGWhiteboard** is a useful feature supporting of the copy&paste of diagram from clipboard. However, we've found the speed for pasting clipboard contents a serious bottleneck for upload and download speed. – **IGFile Sharing** is useful during a discussion, though the upload of the file uses up bandwidth and reduces the whole meeting performance. – **ShareURL** is an useful feature allowing one to post a web page to another users' computer with single click of the button. – **IGChat** not very useful as audio obviates need for text-line chatting.	– Meeting needed to pre-arranged, which is not feasible in many of our experiment scenarios. – Meeting must be conducted within a pre-defined virtual venue. – The only built-in feature is text chat. – No other built-in features unless users install the plug-in themselves as part of AGTk.
(c)	– The bridge servers offered by inSORS and Manchester Computing have been found reliable. – However, the inSORS bridge server offered by Manchester Computing runs version 1 which can results incompatible with inSORS server itself that runs version 2. – Certain audio incompatibility issues have been encountered between the bridge server used inSORS and AGTk.	– The only bridge server offered by Manchester Computing used by AGTk can be unreliable due to high traffic usage.
(d)	– The supports for video and audio hardware is up to date.	– Heavily depends on vic&rat tools, which has not been updated for a few years. – Wide range of hardware has not been supported particularly in audio. No support for USB audio devices.
(e)	– inSORS works behind a firewall as long as the port 554 is opened for port triggering to forward packets to the host AG machine.	– AGTk's performance behind a firewall network can be variable. Commonly, port forwarding of TCP/UDP port 10000-20000 is required to open. – 10,000 port opening can certainly be a serious drawback for experimental sites with tight network security.

(f)	– Education cost of £600 license is required per portal. – Installation is straightforward and configuration found to be easy. After-sales support is considered reliable and good technical services.	– Free to download. – Set-up process can be complicated for novice users. Results after installation process can also be varying between machines and networks. – No technical support apart from a advice from mailing list, which can prolong the whole set-up process.

To expedite our trial, we have purchased a few copies of inSORS for communication between experimental and home broadband sites. The results of using features in inSORS's have been found satisfactory for materials scientists, particularly the features of "Shared Whiteboard" and "File Sharing". The ease of use of IGMeeting in that one can set-up a meeting with other inSORS users without the need for a pre-arranged meeting had been proven a great advantage over AGTk.

4. Analysis and Outcome of Data Management

4.1 Choice of Web Portal (JSR-168 and WSRP)

Whilst the web portal concept can acts as an efficient medium for our "Shared Workspace" (discussion), to download/upload information (data achiving/restore) or even to retrieve previous experiment histories (playback virtual experiments), it is imperative that our web portal conform to a standard to allow efficient ways to deploy, share and even reuse portlets from various other VRE projects. It came to our attention that JSR 168 is a Java Specification Request (JSR) that establishes a standard API for creating portlets.

We have investigated two main JSR-168 compliant web portals, GridSphere [8] and uPortal [3] frameworks and come to the following summary conclusions:

GridSphere

There is a large community who use GridSphere as their main web portal service, but we found the support and reliability of the software failed to live up to expectations given our desire to run and develop on Windows based platforms.
Nonetheless, GridSphere deployment on other LINUX based projects such as GECEM at Cardiff University [9], was successfully achieved, albeit proving challenging at times. We further note that the latest released Java-based framework GridSphere v2.1.4 is more developed towards Linux-based than Windows, making it unstable and failing to even get pass the setup installation stage under the Windows environments. Although the initial setup in Linux environments had been successful, portlet deployment only proved stable if a lower version of Java library and Apache web server was installed on the host machine.

uPortal

This web portal has proven stable under our rapid development of web services under Windows and Linux environments. Its well documented web-site is a valuable resource. Furthermore, the developer's release v3.0 has been modified for the use of JSR-168 as a native module, rather than using conventional Apache's Portlet-to-Channel adapter. This has given us greater ease-of-use and reliability when using this web portal for our development.

Web Service for Remote Portlets (WSRP)

While XML-based web services have been used in different API platforms to transfer data between them, a new concept, Web Service for Remote Portlets (WSRP), allows portlets to be exposed as Web services [10]. The resulting web service will be user-facing and interactive among different web portals. Unlike traditional Web services, WSRP will carry both application and presentation logic that can be displayed by a consuming portal. To the end user, remote portlets will look like and interact with the user just as local portlets would. Unfortunately, WSRF is not yet an agreed standard and therefore interoperation is not guaranteed. Furthermore, it is still at an early stage of development from certain software vendors, it is therefore, difficult to judge and implement a real-pluggable WSRP portlet. Although various .NET 2 web portal libraries have the features to allow implementation of Web Portal service, the use of WSRP has not been implemented to allow interoperability. We are, however, closely monitoring the development of WSRP and our choice to use uPortal library that conforms to the current WSRP standard will enable us to exploit other WSRP modules when it is incorporated within a wider range of software vendors' standard.

Figure 2: Prototype ISME Web Portal

4.2 Web Portlets for Materials Scientists

We have conducted a series of structured interviews with material scientists following the completion of a questionnaire in order to ascertain which web services are required by the material scientists. The interviews have been conducted with members of differing roles and levels of experience, namely, a Professor, a project manager, a lecturer, an instrument scientist and three PhD students. The principal issues/requirements are summarised below:

- Communications concurrent with scene/data visualisation is required. This will help to identify experimental issues.
- The need for a shared desktop/workspace for better collaboration. This is especially necessary to communicate problems and share data.
- A data archive system is required, so that the users are able to retrieve documents and data, to have easy access to previous work, with the experiments systematically recorded.
- An electronic log book of the experiment would be useful, very simple and easy to use, including only pictures and text.
- A catalogue tool to organise the data transfer.
- A tool to ensure the latest version of the data.
- A tool to overcome the problem of sending large datasets (GBytes) back university site – maybe by using a very fast Internet connection.
- AG meeting should be made more user-friendly, easier to set up at any time, reliable, easy to use, portable, as a package, not to require additional time to set-up.
- Access to a powerful resource computer via fast internet to analyse results quickly.
- A project scheduling tool would be also useful to plan the experiment and to keep a diary during experiments.
- Possibility to simulate the experiment in advance, like a virtual experiment, so that new students can get used to the procedure and the facilities.

Based on the above summary outcomes, we had developed a prototype ISME web portal as shown in Figure 2 offering the required services:

- **Error Handling tab**: for debugging purposes during this prototyping phase.
- **News tab:** to display daily news from various news channels provided by uPortal**.**
- **Data Related Services:** Archive Data, File Manager, Catalogue Tool, Virtual Log-Book
  These services relate to data archival and management tasks. The catalogue tool allows materials scientists to manage and save their experimental results in a set of discipline categories of database format. This will allow PhD supervisors or other colleagues to easily access and acquire information when needed. The Virtual Log-Book is more a personal service where experimenters can keep their notes and hourly experimental results digitally on to the centralised server for retrieval when needed in future.
- **Remote Processing Services:** This service aims to provide the material scientists with the ability to connect remotely to their local machine back at the home university site and to launch different useful application installed there, such as

ABAQUS, ImageJ, GSAS, OpenGini, Exact, FORTRAN, MS Excel, MS Word etc. This service ensures that the user is properly authenticated into the remote host and gives them the opportunity to use the software applications they usually employ to visualise data, analyse results, make conclusions etc.

- **Communication:** Image, Video, Chat, Shared Desktop, Message Board, AccessGrid.
  This service allows materials scientists to communicate to each other, for example, sharing images and video on the web portal. A common shared workspace with AG tools can be used to visualise results from the experiments via the multimedia resource, whereby the whole team can 'meet' together bringing their own data, modelling predictions and discuss and develop an evolving experimental strategy.
- **Miscellaneous:** Calendar, Project Scheduling, Virtual Experiments, Email
  The calendar service is to support experimental scheduling and also allows PhD supervisors or co-workers to locate specific experiments of a materials scientist. Virtual Experiments can be undertaken using a legacy application developed by the Open University [11].
  This service aims to provide the researchers with a simulation tool of the experiments at the remote site, which they can use prior their visit to the facility to learn the experimental process or even on-site in case any problems arise.

We have shown the above prototype web portal to materials scientists for initial feedback. The materials scientists perform different roles at the university and during the experiments. The analysis of the feedback results showed that the wide range of web services was appreciated and that no services were perceived as lacking.

There are other services which can be improved and the concepts of these services need to be clear so as not to replicate any of the currently available applications. For example,

- Virtual Log Book would require time stamps on the experiments as an automatic feature to provide an historical record of actions.
- Archive Tools need to be merged with the Catalogue tool to reduce replication within the web portal.
- The Calendar service must be linked to experiments allocated by the experiment al facility and should made public to allow other colleagues to view timetables.

Some overlap between services has been identified in this phase as well as some irrelevant services for the experimenters such as Chat, Email and Video.

5. Future Work

We are closely collaborating with a wide-range of material scientist users to determine and implement our final phase of the ISME web portal. Feedbacks from the first phase had given us the information to prioratise and implement the services required by the materials scientists. Technical lessons learnt from the uPortal will also be documented as we believe this maybe of interest to the VRE community.

At the same time, we are collaborating with Manchester Computing to develop an AG Meeting Notifications tools (based on vic&rat) to allow meetings to be conducted instantly. This open source, licence-free and cost-free customisable AG Meeting Notification tools will allow greater use within VRE and materials scientist communities.

We have a Pan-Tilt-Zoom (PTZ) camera installed in the experimental hutch to allow remote users to control the static viewpoint. In addition we have also developed a novel AG viewpoint: the Mobile-AG node. The Mobile -AG gives remote users a first person viewpoint and improved insight regarding the mounting and positioning of a materials sample. It uses a mobile camera system attached to the experimentalist's glasses so that their operations in the experimental hutch can be broadcast to scientists at remote sites. This gives remote team members an insight into the local environment in a more flexible manner than is available with static cameras. This allows us to share views not otherwise available and may help us to diagnose experimental problems with the equipment, carry out repairs under instruction or to allow remote users to offer advice about the experimental set-up. At this stage of the project, we are still investigating the full potential use of Mobile-AG used in the experiment hutch.

In summary, this project focuses on understanding how VRE tools can assist large multi-site teams undertake experiments at international facilities. Our aim is to exploit and customise existing tools in such a way that scientist can use them seamlessly and effectively in undertaking experiments. So far we have undertaken a small pilot study based around two experimental facilities and one extended research team. This project will be expanded to a wider range of users with assessment criteria needed for evaluate the prototype portal and services. We will have an opportunity in July 2006 to have a full test-drive of our AG tools and Web portal on the Daresbury 16.3 beamline using public access users, since this is to become a public service beamline maintained partly by the Manchester team. If this proves successful we will examine the possibility of transferring the approach to the new Diamond Light Source near Oxford. This has created a unique opportunity to embed and test our VRE tools with a large community of new users to assess how these tools support and impact the learning curve of new users.

6. References

1. K.T.W. Tan, D. Tsaneva, M. Daley, N.J. Avis and P.J. Withers, Advanced Collaborative Tools for Engineering Body Scanners. UK e-Science Programme All Hands Meeting, 2005, Nottingham
2. K.T.W. Tan, N.J. Avis, G. Johnson and P.J. Withers, 2004, Towards a grid enabled engineering body scanner. UK e-Science Programme All Hands Meeting 2004, Nottingham
3. uPortal – http://www.uportal.org
4. inSORS – Multimedia Conferencing & Collaboration Software, http://www.insors.com
5. The Access Grid Project – a grid community, http://www.accessgrid.org
6. Vic & Rat – http://www-mice.cs.ucl.ac.uk/multimedia/software/
7. Virtual Room Videoconferencing System – http://www.vrvs.org
8. GridSphere Portal – http://www.gridsphere.org
9. M. Lin, D.W.Walker, Y. Chen and J.W. Jones, A Web Service Architecture for GECEM, All-Hands Meeting, 2004, Nottingham
10. Web Services for Remote Portlets – http://www.oasis-open.org/committees/wsrp
11. J. A. James, J.R. Santistiban, M.R. Daymond and L. Edwards, Use of a Virtual Laboratory to plan, execute and analyse Neutron Strain Scanning experiments, NOBUGS 2002

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

DyVOSE Project: Experiences in Applying Privilege Management Infrastructures

J. Watt, J. Koetsier, R.O. Sinnott, A.J. Stell

National e-Science Centre, University of Glasgow
jwatt@nesc.gla.ac.uk

Abstract

Privilege Management Infrastructures (PMI) are emerging as a necessary alternative to authorization through Access Control Lists (ACL) as the need for finer grained security on the Grid increases in numerous domains. The 2-year JISC funded DyVOSE Project has investigated applying PMIs within an e-Science education context. This has involved establishing a Grid Computing module as part of Glasgow University's Advanced MSc degree in Computing Science. A laboratory infrastructure was built for the students realising a PMI with the PERMIS software, to protect Grid Services they created.. The first year of the course centered on building a static PMI at Glasgow. The second year extended this to allow dynamic attribute delegation between Glasgow and Edinburgh to support dynamic establishment of fine grained authorization based virtual organizations across multiple institutions. This dynamic delegation was implemented using the DIS (Delegation Issuing) Web Service supplied by the University of Kent. This paper describes the experiences and lessons learned from setting up and applying the advanced Grid authorization infrastructure within the Grid Computing course, focusing primarily on the second year and the dynamic virtual organisation setup between Glasgow and Edinburgh.

1. Project Background

The DyVOSE Project (Dynamic Virtual Organisations in e-Science Education) is a JISC funded two-year project investigating the establishment of a Privilege Management Infrastructure (PMI) that supports dynamic delegation of authority in the context of a Grid Computing Advanced MSc. module at the University of Glasgow. Specifically the project is investigating the application of the PERMIS software in creating an attribute management infrastructure that allows institutions to establish trust relationships that will assert and enforce the privileges presented by attributes issued by external institutions.

In the first year of the project a static PMI was implemented using the PERMIS authorization function. This allows two teams of students to author their own GT3.3 services and restrict access to certain methods provided the student held the appropriate 'team' attribute. In this case, all privileges were issued by Glasgow so no cross-organisational infrastructure was necessary [1].

In the second year the students created a GT3.3 service which ran a BLAST [2] query against a set of data retrieved from a data store hosted at Edinburgh University. Students were again split into two teams, one running a query against nucleotide data and one against protein data. PERMIS was used to secure the services at both sides, denying access to students in the protein team who attempted to extract and match nucleotide data and vice versa. In this scenario, inter-institution interaction was required, so user attributes needed to be recognized at both institutions. This may be implemented statically in the same way as the first year assignment by completely sharing user information between sites, but this is highly undesirable if we wish to deploy this kind of setup using existing campus directories. A more scalable and realistic Grid model is where local sites maintain information on their *own users* and define their own local security policies restricting access to local resources by both local users and trusted remote users/sites. The Delegation Issuing Service (DIS) aims to provide a safe, intuitive environment in which institutions may establish chains of trust without surrendering sensitive local user information. The fundamental benefit of the DIS with regard to Grids is to support fine grained authorization infrastructures whereby attributes needed for a given virtual organization can be dynamically created and recognized by remote "trusted" sources of authority. Through this model, virtual

organizations can be created in principle "on-the-fly" without detailed agreements.

2. PMI Technologies

A number of authorisation control mechanisms exist that can be applied to the Grid, examples of these include CAS [3], Akenti [4] and VOMS [5]. Each offer their own advantages and disadvantages [6]. PERMIS (Privilege and Role Management Infrastructure Standards Validation) [7] is a Role Based Access Control System [8] which uses X509 Attribute Certificates (ACs) [9] to issue privileges to users on a system. VOMS provides ACs to its users, but attributes are still handled from a central server. PERMIS is completely decentralised and access control decisions are made locally at the resource side. These access control decisions are typically made through attribute certificates (ACs) signed by a party trusted by the local resource provider. This might be the local source of authority (SoA), however a more scalable model is to delegate this responsibility in a strictly controlled manner to trusted personnel involved in the virtual organization (VO).

In a Grid context it is unrealistic to expect all information to be maintained centrally. VOs may well have many users and resources across multiple sites, and these users come and go throughout the course of the VO collaboration. Knowing for example that a given user is at Glasgow University is best answered by the Glasgow University authentication processes. However, whilst knowledge of a given users status may well be best answered by that users home authentication infrastructure, the roles and responsibilities needed to access remote resources specific to that VO may best be delegated to trusted personnel associated with that VO – it is this capability that the DIS service is to support. To achieve this requires that an authorization infrastructure exists that can firstly define appropriate policy enforcement (PEP) and policy decision points (PDP), i.e. define and enforce the rules needed to grant/deny access requests based upon ACs .

PERMIS offers a generic API which can be applied to any resource, so our investigations could also be applied to non-Grid regimes.

The PERMIS decision function can be a standalone Java API, or it can be deployed in the same container as the Grid Service it is intended to protect. The GGF SAML AuthZ API [10][11] provides a method for Globus to bypass the generic GSI [12] access control and allow external services to make authorisation decisions. Once deployed, the PERMIS service requires an XML policy which describes in complete detail the targets, actions and roles which apply at the resource (or at the institution). This policy may be written using the Policy Editor GUI supplied with the PERMIS software, or may be edited by hand. Another important GUI supplied with PERMIS is the Privilege Allocator (or the slightly more user friendly Attribute Certificate Manager (ACM)). This is responsible for allocating roles and signing ACs for users. This tool can also be used to browse LDAP directories for ACs and can be useful in confirming that ACs have been loaded correctly.

In order to implement a dynamic PMI, extensions to this ACM tool need to be made. As it stands, the ACM can issue any certificate it wishes, irrespective of its validity within the PMI. Ideally a method of enforcing the infrastructure described in the XML Policy to allow an administrator to only be allowed to issue valid ACs is needed. To keep user information at the home site, it would be necessary to have a mechanism that would allow a remote administrator to issue ACs only with roles relevant at their institution to the home site LDAP. There are two gains to this, the first being that the remote admin can only operate within a very restricted attribute set, which would exclude any possibility of them issuing home site roles. The second gain is that, as requested, all important user data is still held at the home institution.

The Delegation Issuing Service (DIS) is intended to provide this functionality. The DyVOSE project is the first project to investigate this technology to any great extent, with the goal of providing a user and admin guide to the installation and operation of the DIS service. In the next section we describe some of the technicalities of setting up and using the DIS and then we outline how we have applied this technology for teaching purposes.

3. The Delegation Issuing Service

In the current implementation, the DIS software is a web service consisting of a Java library based on the Tomcat and AXIS SOAP servers. The web service is accessed by a DIS client written in PHP running on an Apache server which acts as a proxy between the DIS service and the user. This client invokes the Java component through SOAP calls, and is presented to the user in a web browser after mandatory authentication to Apache using their own username and password. These components may be hosted on separate

machines, although for the DyVOSE investigations, they were situated on the same computer.

The DIS service assumes that there is a Tomcat application server of recommended version 4.x installed on the server side, along with an LDAP server for attribute storage and Apache authentication. A Java Runtime Environment of at least version 1.4 is required. On the client side, a functional Apache web server loaded with the SSL, PHP and LDAP modules is required. The Apache and Tomcat servers were hosted on the same server with no incompatibility issues encountered. The resource OS was chosen to be Fedora Core 4 as this distribution contained all the Apache functionality and extra modules as standard RPMs and were loaded (and to some extent, configured) automatically. Edinburgh has successfully (in terms of providing default functionality) migrated the DIS service to Fedora Core 5. In addition, the LDAP backend (Berkeley DB) was of a version advanced enough to allow the most recent version of OpenLDAP to be installed on the machine. For the purposes of the Grid Course assignment at Glasgow, this was abandoned in favour of a slightly older version of LDAP which was compatible with the GT3.3 PERMIS Authz service deployed on a separate machine, although the Edinburgh DIS was successfully integrated with the newest version of LDAP.

The DIS software itself ships as two gzip files containing the server and client side tools separately. These files include the necessary Java libraries, Web Service Deployment Descriptor file, LDAP core schema, and configuration files. A sample LDIF file containing the required DIS users and their certificates is provided for loading the LDAP server to test the installation. Due to the complexity of the surrounding PKI, this file is essential for installation as it is HIGHLY unlikely that a DIS-friendly certificate infrastructure could be deployed prior to confirming the success of the DIS install. One drawback with this is that using this file forces the DIS service to handle users with a fixed Distinguished Name (DN) which makes that particular setup quite non-portable.

The implementation of the DIS requires a consistent PKI comprising a total of around 9 certificates and key pairs in order to realise the service and its proxy. In several cases, the certificates need to be loaded in three different formats (PEM, DER and p12) [13] in order to talk to the various components of the DIS service, and the underlying PERMIS server that the DIS creates ACs for. These certificates are created from the command line using *openssl* after creation of a site specific configuration file which handles certificate extensions and populates the DN with a structure corresponding to the users present in LDAP. The two key users in the DIS infrastructure are the Source of Authority (SoA) and the DIS user. These are the only two users who require pre-loaded ACs as everyone else in the LDAP server can be allocated ACs by the DIS. The AC is stored as an attribute labelled *attributeCertificateAttribute*, with an optional *;binary* extension dependent on the local LDAP schema. The AC is created using the PERMIS Attribute Certificate Manager (ACM) GUI, which can also load the certificate into LDAP. These two pre-loaded ACs are essential to the operation of the PMI, and for the SOA and DIS user, they contain an attribute *'permisRole'* whose entries are a list of all assignable roles within the infrastructure. In the case of the DIS, the attribute list is an explicit statement of every role that the DIS can assign and delegate. In addition to the attribute list, the SOA requires another attribute called *'XML policy'* which contains the XML file representing the site policy. This policy states the hierarchical relationships between roles, which targets and actions these roles apply to, the scope references, and which SOAs are to be trusted within the VO.

The SoA is the root of trust for the PKI, and signs every certificate (and through the DIS, every Attribute Certificate) within the infrastructure. A PEM format root certificate was created using *openssl*, along with its corresponding encrypted private key. This certificate is required to be present in the Apache SSL configuration and is used to create user certificates compatible with the Globus Toolkit, allowing Grid users to interact with the PMI and be assigned meaningful privileges. The root certificate is required to be loaded into the SOA node of the LDAP server, in this case the PEM certificate needs to be converted to the DER format using openssl. In LDAP, this certificate is loaded under the *"userCertificate"* attribute, again with an optional *";binary"* suffix depending on the version of LDAP. In addition, this DER format file is required by the DIS server for validation, and also needs to be loaded into the Tomcat server keystore and the Java JRE security CA keystore. Finally, the root certificate and its private key need to be converted to a PKCS p12 format which is used by the Attribute Certificate Manager (ACM) to sign the SoA ACs.

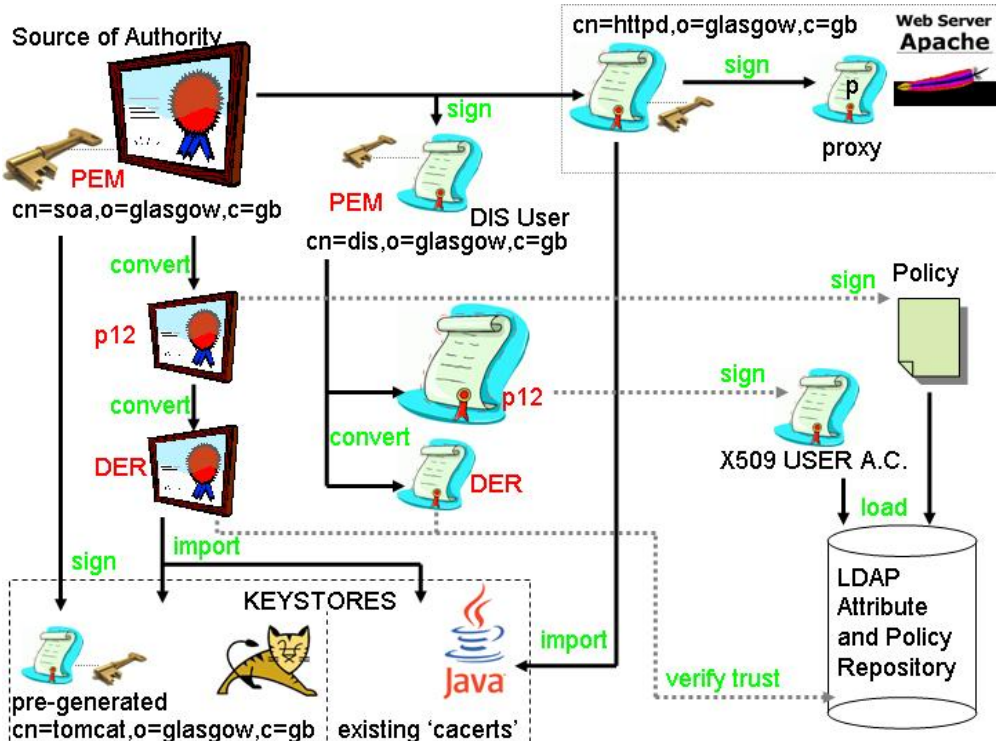


Figure 1: A schematic of the underlying PKI within the DIS service.

The DIS user is the other key entity within the PMI, and it is this user who is responsible for signing ACs and assigning and revoking privileges. In the same way as the SoA, the DIS user is assigned a certificate matching its LDAP DN, and this certificate is converted to DER format and loaded into LDAP using the '*userCertificate'* attribute as above. This is to complete the validation chain for the PERMIS server when it attempts to verify the credentials of a presented user certificate. The DIS certificate is converted to PKCS12 format and loaded into the Tomcat configuration directory.

Two keystores need to be maintained within the DIS service. The generic Java JRE security CA trust keystore needs to be loaded with the root certificate (in DER format). A second keystore for the Tomcat server needs to be created using the Java keytool. A private key and certificate for the tomcat server is created and loaded into the keystore, along with the root certificate (DER) again to verify the certificate. A schematic of the Public Key Infrastructure required to realise a basic DIS service is shown in Figure 1.

The DIS service is operated by logging in to the DIS proxy which retrieves its authentication information from LDAP. Any ACs issued by this service are always signed by the DIS user, as it is his certificate (p12 file) which is in the signing certificate repository. However, information on the user that logs in will determine which roles can be assigned or delegated. The SoA has the ability to delegate and grant all privileges within the PMI, but delegated users, including the DIS, will only be able to delegate those roles that they possess and are authorised to delegate. In the same way, the DIS web service will only sign certificates and issue roles if the user logging into the DIS possess the relevant credentials and is authorised to delegate those roles..

To understand how this infrastructure facilitates dynamic establishment of security focused virtual organisations, we consider the Grid programming assignment at Glasgow University.

4. Implementation Scenario

As part of the Grid Course programming assignment, students at Glasgow authored a GT3.3 Grid Service which used BLAST to do similarity searching on a set of either protein or nucleotide data retrieved from a GT3.3 service based at the National e-Science Centre at the University of Edinburgh. These services were PERMIS protected to only grant access to students in the same group (Protein Team P or Nucleotide Team N) provided they presented an AC with the correct role. Initially, these ACs were granted in advance and stored at both

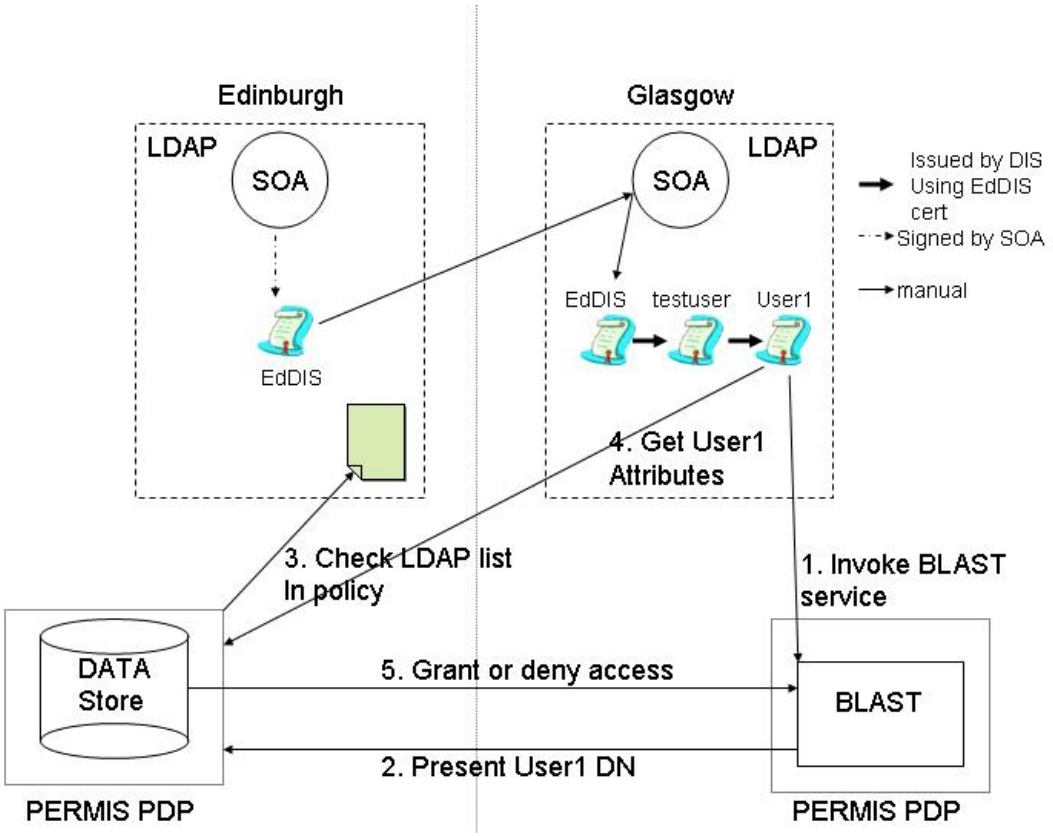


Figure 2: Diagram of the interactions required for Edinburgh to issue an AC granting access to its resources.

locations for testing the service functionality. This is undesirable as in general, information on students should only really be present at the student's home institution. The benefit of this approach is that the implementation of this static PMI is well understood and easily maintained through expertise gained in the previous year's work.

To extend this static PMI to a model supporting dynamic delegation, a DIS service was created at both sites. Using the Glasgow DIS, the Edinburgh SOA could login and grant Glasgow users the privilege to access the Edinburgh data store. This way Glasgow retains all its student's details, yet through privilege delegation, an Edinburgh user can grant these users a role recognised by Edinburgh provided they have been granted this privilege by the Glasgow DIS.

A number of different approaches were considered based on the current version of the PERMIS software. One method which was attempted was the use of LDAP referral, which would allow a single LDAP to be specified by the Edinburgh PERMIS Policy Decision Point (PDP) for it to retrieve its user attributes from. With referral set up, a branch of the Edinburgh LDAP server would point to the Glasgow LDAP as if it were part of its own tree. This approach ran into several problems, the main one being that when the PERMIS PDP searched the LDAP tree and came across a referral, the LDAP server bounced the details back to the PDP for it to do the search itself on the remote LDAP. Since no functionality exists for this the PDP crashed each time it encountered this referral. Attempts to make this referral transparent to the PDP, i.e. getting the LDAP server to do the retrieval and presenting remote attributes as if they came from the local LDAP were not successful. The PERMIS team assured us that the PDP LDAP server parameter could take several values, however despite numerous attempts this was never made to work. A solution was found in which multiple LDAP servers could be listed within the site policy itself under a "Repository Policy" subject tag. This method meant that referral was not necessary nor did it compromise local security since the entry was merely a location in which to look for attributes, and not a statement of trust on the part of the local SoA.

Two aspects of dynamic delegation which at the time of writing were not implemented in the PERMIS software were those of role mapping and authority recognition. Role mapping allows separate sites with their own security policies to

state which roles that apply at one site can match the roles at another site. Typically, an institution will define "External" roles that have lower privilege than the local ones and these roles are typically used as equivalences. Since this functionality was not present on implementation, it was forced upon the PMI by an agreement which stated that the external roles at both institutions would be given exactly the same name. Now the only difference between an external Glasgow role and an external Edinburgh role is that of which SoA (or in this case, DIS) actually signed the user's AC.

The second function which was not available yet was that of recognition of authority, or how the VO formed between the two sites would recognise ACs signed by the other site. An easy solution to this is to add the external SoA to the "SoA Policy" tag within the XML policy. This way, any ACs extracted which have been signed by the remote host site can be verified. This method, although easy, means that a given site explicitly trusts all of the actions of the remote SoA. Without a DIS service protecting the assignment of ACs, the remote SoA could in principle assign any role they are aware about from the home institution to any of its users. The DIS service, since it only ever issues valid ACs within the constraints of its own site policy, can enforce more stringent rules on what the remote SoA can allocate. However, we suggest a different approach which has been implemented in our dynamic delegation scenario.

Instead of trusting an external SoA to establish a chain of trust, we created a *EdDIS* user at the remote host who has been allocated a DIS certificate containing all the roles they may delegate, but which has been signed by the home institution. Therefore when a Glasgow user presents an AC which has been signed by this *EdDIS* user (within the Glasgow DIS) this AC will already be trusted by Edinburgh.

To understand this we provide an example of granting access to the Edinburgh data to a Glasgow user. This sequence is shown pictorially in Figure 2, with the PERMIS decisions on the Glasgow side being omitted as this is a purely static PMI function.

The Edinburgh SoA creates an "*EdDIS*" signing key pair which is signed by itself. This certificate is handed to the Glasgow SoA (via a secure channel) and the administrators on the Glasgow side mount this certificate in their LDAP directory. The Edinburgh SoA also creates an AC, issuing two external roles and the ability to delegate those roles, to this user, "*GlaStudentTeamN*" and "*GlaStudentTeamP*. Now an extra user *EdDIS* appears in the Glasgow DIS. To demonstrate delegation, a new user called "*testuser*" was created at Glasgow, who was issued with an AC signed by *EdDIS* which allowed the user to delegate the external roles to other Glasgow users (in this case "*User1*"). The "testuser" can then log into the Glasgow DIS and create an AC for User1 containing the role "*GlaStudentTeamN*".

User1 calls the Edinburgh Grid Service through their own Glasgow BLAST service. GSI passes User1's DN to the Edinburgh PERMIS PDP. The PDP reads the Edinburgh policy, and locates the LDAP server to extract the User1 attributes (contained in the Repository Policy list). The remote LDAP is queried, and the *User1* AC is extracted. The PDP checks the signature on the AC and verifies that it has been signed by the *EdDIS* user, who although existing at Glasgow, is signed by the Edinburgh SoA. Since the chain of trust can be verified back to the Edinburgh SOA, who is the trusted SoA in their policy, the PDP establishes this is a valid AC. Then the PDP makes the decision based on the user attribute presented whether to release the Protein or Nucleotide data to the Glasgow Grid Service. Any ACs, that are part of the Glasgow PMI, issued by the Glasgow SOA, will simply be ignored by the Edinburgh PDP, as they are not signed by any party that the Edinburgh SOA trusts. This allows two PMIs to co-exist in one LDAP tree.

The DIS can assign and revoke user ACs as many times as it wishes, without affecting any other user certificates in its infrastructure. Also, any users who have delegated their roles to other people will find that the roles they delegated will still be valid even if they cease to be members of that PMI. A screenshot of the DIS service window is shown in Figure 3.

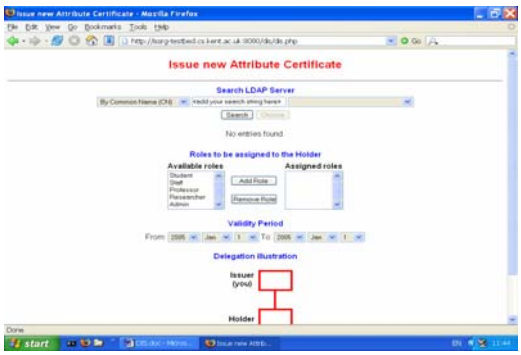


Figure 3: The DIS service Web Interface

5. Experiences and Conclusions

The DIS software shows great promise as a tool to enable dynamic VO establishment. We have successfully demonstrated a VO which allows a SoA at a remote site to securely assign and revoke privileges to home users, without user information being factored out to external databases. The adherence of the DIS service to the local policy means that only valid ACs can be issued, and the scenario described above allows the establishment of distributed trust without surrendering local security. Once installed the service is intuitive, with a GUI interface that allows AC issuing in a few seconds.

The Grid Computing module is now completed. Of the 11 students that took this module this year, all but one managed to access and retrieve data from the PERMIS protected service in Edinburgh, thus providing that the infrastructure works.

The work has not been without issues however. The lack of availability of source code due to commercial concerns makes the tracking of errors and diagnosis of problems very problematic. This extended the development time of this project by an unacceptable amount due to our reliance on the tireless help of the PERMIS development team, in particular Sassa Otenko whom we are grateful to for his efforts. The underlying PKI to establish the DIS service is over-complicated, most certificates are required to be duplicated and converted many times through the system. This is probably due to the Web Proxy based approach of the GUI, which demands many certificates and keystore entries to be maintained. Some of our scenario definitions have had to change on several occasions due to undocumented features within the DIS and PERMIS.

In the absence of source code, some heavier documentation would be desirable, in particular with regard to setting up the PKI. A simpleCA approach that could generate the appropriate keys, in the appropriate formats according to the domain structure of your institution would be invaluable. Once running, the software is easy to use and robust, but the implementation time required at this stage of the software development may be outside the remit of any developers who are new to this technology.

Nevertheless the proof of concept that dynamic delegation of authority works has been a major output of the project. We believe that this federated model of policy definition and management that suit the needs of a multitude of VOs has numerous potential application areas. One key area of focus is to support and extend the largely static nature of ACs used for example in Shibboleth identity and service provider interactions. Shibboleth access to resources has up to now been based largely upon an agreed set of attributes and their values based for example around the eduPerson object class (www.eduperson.org). This model is not conducive to the more dynamic nature of short lived Grid based VOs which come together for a given time to solve a particular problem. In this case, dynamic creation and recognition of attributes based on a limited trust model is more apposite. As such, we plan to explore this technology in a range of other e-Science projects at the National e-Science Centre in Glasgow.

5.1 Acknowledgements

The DyVOSE project was funded by a grant from the Joint Information Systems Committee (JISC) as part of the Core Middleware Technology Development Programme. The authors would like to thank the programme manager Nicole Harris and collaborators in the project. In particular special thanks are given to Professor David Chadwick and especially Dr Sassa Otenko for help in exploring the DIS and PERMIS technologies.

6. References

[1] R.O.Sinnott, A.J.Stell, J.Watt, “Experiences in Teaching Grid Computing to Advanced Level Students” Proceedings of CLAG+GridEdu Conference, May 2005, Cardiff, Wales

[2] BLAST (Basic Local Alignment Search Tool), http://www.ncbi.nih.gov/Education/BLASTinfo/information3.html

[3] L. Pearlman, et al., “A Community Authorization Service for Group Collaboration” in Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002

[4] Johnston, W., Mudumbai, S., Thompson, M., “Authorization and Attribute Certificates for Widely Distributed Access Control”, IEEE 7th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Stanford, CA, June 1998, p340-345 (http://www-itg.lbl.gov/security/Akenti)

[5] VOMS Architecture, European Datagrid Authorization Working Group, 5th September 2002

[6] A.J.Stell, “Grid Security: An Evaluation of Authorisation Infrastructures for Grid Computing” MSc Dissertation, University of Glasgow 2004

[7] Privilege and Role Management Infrastructure Standards Validation project (www.permis.org)
[8] D.W. Chadwick, O. Otenko, "The PERMIS X509 Role Based Privilege Management Infrastructure", Future Generation Computer Systems, 936 (2002) 1-13, December 2002, Elsevier Science BV
[9] D.W.Chadwick, O. Otenko, E.Ball, "Role Based Access Control with X.509 Attribute Certificates", IEEE Internet Computing, Mar-April 2003, pp. 62-69
[10] V. Welch, F Siebenlist, D.Chadwick, S. Meder, L. Pearlman, "Use of SAML for OGSA Authorization", June 2004, https://forge.gridforum.org/projects/ogsa-authz
[11] OASIS, Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1, 2 September 2003, http://www.oasis-open.org/committees/security
[12] Globus Security Infrastructure (GSI) http://www.globus.org/security
[13] OpenSSL: http://www.flatmtn.com/computer/Linux-SSLCertificates.html

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Building a Modular Authorization Infrastructure

David Chadwick, Gansen Zhao, Sassa Otenko, Romain Laborde, Linying Su, Tuan Anh Nguyen

University of Kent

Abstract

Authorization infrastructures manage privileges and render access control decisions, allowing applications to adjust their behavior according to the privileges allocated to users. This paper describes the PERMIS role based authorization infrastructure along with its conceptual authorisation, access control, and trust models. PERMIS has the novel concept of a credential validation service, which verifies a user's credentials prior to access control decision making and enables the distributed management of credentials. Details of the design and the implementation of PERMIS are presented along with details of its integration with Globus Toolkit, Shibboleth and GridShib. A comparison of PERMIS with other authorization and access control implementations is given, along with our plans for the future.

1. Introduction

Authorization infrastructures provide facilities to manage privileges, render access control decisions, and process the related information. Normally, an authorization infrastructure will follow a certain set of authorization policies to make decisions, such as Credential Issuing Policies, Access Control Policies, Delegation Policies, and Credential Validation Policies. These policies contain the rules and criteria that specify how privileges (or credentials) are managed and access control decisions are made. Following these policies, an authorization infrastructure issues credentials to users, who might belong to one domain, whilst the credentials are then presented and validated by the authorization infrastructure of the resource which might belong to a different domain. Once the credentials are validated, the authorization infrastructure then renders an access control decision, and returns this to the application for enforcement. When enforcing the access control decisions, the application should only grant those requests that are authorized by the authorization infrastructure, and should forbid all others.

The authorization infrastructure that we have built is called PERMIS [1]. This paper describes the various components of the PERMIS authorization infrastructure, the conceptual models that are behind them, and the standards that we have used. We also describe some of our plans for future enhancements and work that still needs to be done. We also compare our work to that of others. The rest of this paper is structured as follows. Section 2 provides the conceptual models of our authorization infrastructure. Section 3 describes the design and implementation of PERMIS. Section 4 presents PERMIS's integration with Globus Toolkit, Shibboleth and GridShib. Section 5 compares PERMIS to other related research. Section 6 concludes and indicates our plans for the future.

2. Conceptual Models

2.1 The Authorisation Model

The authorization model paradigm adopted is the "Subject – Action – Target" paradigm and the Role Based Access Control model [18], where roles are presented as credentials issued to the subjects. In our model, a role is not restricted to an organizational role, but can be any attribute of the subject, such as a professional qualification or their current level of authentication [23]. Each subject represents a real world principal, which is the action performer. Action is the operation that is requested to be performed on the target. It can be either a simple operation, or a bundle of complex operations that is provided as an integrated set. Target is the object of the action, over which the action is to be performed. A target represents one or more critical resources that need to be protected from unauthorized access.

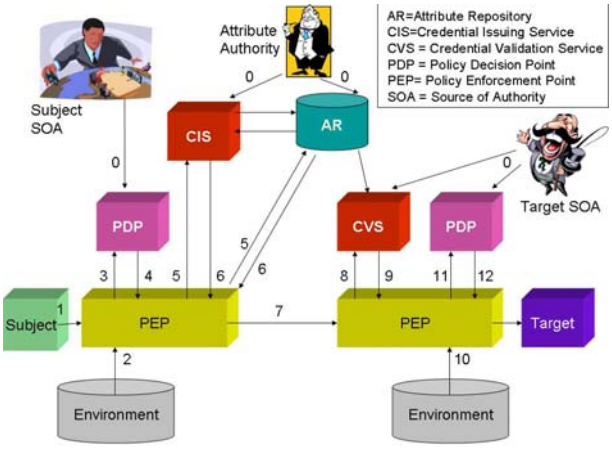


Figure 1: High Level Conceptual Model of an Authorisation Infrastructure

Figure 1 shows our high level conceptual model for an authorization infrastructure. Step 0 is the initialization step for the infrastructure, when the policies are created and stored in the various components. Each subject may possess a set of credentials from many different Attribute Authorities (AAs), that may be pre-issued, long lived and stored in a repository or short lived and issued on demand, according to their Credential Issuing Policies. The Subject Source of Authority (SOA) dictates which of these credentials can leave the subject domain for each target domain. When a subject issues an application request (step 1), the policy decision point (PDP) informs the application's policy enforcement point (PEP) which credentials to include with the user's request (steps 3-4). These are then collected from the Credential Issuing Service (CIS) or Attribute Repository by the PEP (steps 5-6). The user's request is transferred to the target site (step 7) where the target SOA has already initialized the Credential Validation Policy that says which credentials from which issuing AAs are trusted by the target site, and the Access Control policy that says which privileges are given to which attributes. The user's credentials are first validated (steps 8-9) and then the validated attributes, along with any environmental information, such as current date and time (step 10), are passed to the PDP for an access control decision (steps 11-12). If the decision is granted the user's request is allowed by the PEP, otherwise it is rejected. In more sophisticated systems there may be a chain of PDPs called by the PEP, in which case each PDP may return granted, denied or don't know; the latter response allowing the PEP to call the next PDP in the chain.

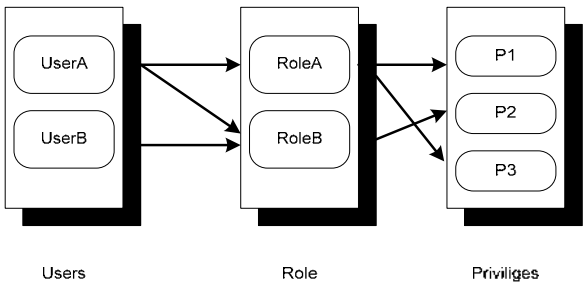


Figure 2 : Example User-Role-Privilege Assignments

PERMIS uses the Role Based Access Control Model [18]. Roles are used to model organization roles, user groups, or any attribute of the user. Subjects are assigned attributes, or role memberships. A subject can be the member of zero, one or multiple roles at the same time. Conversely, a role can have zero, one or more subject occupants at the same time.

Privileges are directly allocated to roles or assigned attributes. Thus each role (or assigned attribute) is associated with a set of privileges, representing the authorised rights the role/attribute has been given by the system administrator. These are rights to perform operations on target objects. Thus a subject is authorised to perform the operations corresponding to his role memberships (or attribute assignments). Changing the privileges allocated to a role/attribute will affect all subjects who are members of the role or who have the assigned attribute.

Figure 2 shows that UserA is a member of both RoleA and RoleB, and UserB is a member of RoleB. RoleA has been granted privileges P1 and P3, whilst RoleB has been granted privilege P2. With Role Based Access Controls this means that UserA is granted privileges P1, P2 and P3 whilst UserB only has privilege P2.

PERMIS supports hierarchical RBAC in which roles (or attributes) are organized in a hierarchy, with some being superior to others. A superior role inherits all the privileges allocated to its subordinate roles. For example, if the role Staff is subordinate to Manager, the Manager role will inherit the privileges allocated to the Staff role. A member of the Manager role can perform operations explicitly authorized to Managers as well as operations authorised to Staff. The inheritance of privileges from subordinate roles is recursive, thus a role r_o will inherit privileges from all its direct subordinate roles r_s, and indirect subordinate roles which are direct or indirect subordinate roles of r_s.

2.2 The Trust Model

Credentials are the format used to securely transfer a subject's attributes/roles from the Attribute Authority to the recipient. They are also known as attribute assertions [20]. PERMIS only trusts valid credentials. A valid credential is one that has been issued by a trusted AA or his delegate in accordance with the current authorization policies (Issuing, Validation and Delegation policies).

It is important to recognize the difference between an authentic credential and a valid credential. An authentic credential is one that has been received exactly as it was originally issued by the AA. It has not been tampered with or modified. Its digital signature, if present, is intact and validates as trustworthy by the underlying PKI, meaning that the AA's signing key has not been compromised, i.e. his public key (certificate) is still valid. A valid credential on the other hand is an authentic credential that has been issued according to the prevailing authorization policies. In order to clarify the difference, an example is the paper money issued by the makers of the game Monopoly. This money is authentic, since it has been issued by the makers of Monopoly. The money is also valid for buying houses on Mayfair in the game of Monopoly. However, the money is not valid if taken to the local supermarket because their policy does not recognize the makers of Monopoly as a trusted AA for issuing money.

Recognition of trusted AAs is part of PERMIS's Credential Validation Policy. PERMIS checks that the AA is mentioned in this policy directly, or that the credential issuer has been delegated a privilege by a trusted issuer, recursively (i.e. a recursive chain of trusted issuers is established controlled by the Delegation Policies of the Target SOA and the intermediate AAs in the chain). The PERMIS Credential Validation Policy contains rules that govern which attributes different AAs are trusted to issue, along with a Delegation Policy for each AA. These rules

separate AAs into different groups and assign them different rights to issue different attributes to different sets of subjects. Further each AA will have its own Credential Issuing Policy and Delegation Policy. PERMIS assumes that if a credential has been issued and signed by an AA, then it must be conformant to the AA's Issuing Policy, so this need not be checked any further. However, if the credential was subsequently delegated this may or may not have conformed to the original AA's Delegation Policy. Therefore when PERMIS validates a credential it checks that it conforms to the AA's delegation policy as well as the Target SOA's delegation policy. PERMIS also makes sure that all credentials conform to the delegation paradigm that an issuer cannot delegate more privileges than he has himself, to ensure constrained propagation of privileges from issuers to subjects.

The net result of this trust model is that PERMIS can support multiple AAs issuing different sets of attributes to different groups of users, in which each AA can have different delegation policies, yet the target SOA can specify an overall Credential Validation Policy that constrains which of these (delegated) credentials are trusted to be used to access the resources under his control.

3. PERMIS: A Modular Authorization Infrastructure

The PERMIS authorization infrastructure is shown in Figure 3. The PERMIS authorisation infrastructure provides facilities for policy management, credential management, credential validation and access control decision making.

3.1 Policy Management

PERMIS Policies are rules and criteria that the decision making process uses to render decisions. It mainly contains two categories of rules, trust related rules (Credential Validation Policy) and privilege related rules (Access Control Policy). Trust related rules specify the system's trust in the distributed Attribute Authorities. Only credentials issued by trusted AAs within their authority will be accepted. Privilege related rules specify the domains of targets, the role hierarchies, the privileges assigned to each role and the conditions under which these privileges may be used, for example, the times of day or the maximum amount of a resource that may be requested.

PERMIS provides a policy composing tool, the Policy Editor [13], which users can use to compose and edit PERMIS policies. The GUI interface of the Policy Editor comprises: the subject (user) policy window, the trusted AA policy window, the role assignment policy window, the role hierarchy policy window, the target resource policy window, the action policy window and the role-privilege policy window. These windows provide forms for users to fill in, then the tool generates the corresponding PERMIS policy. Policies can be saved in pure XML format, or the XML can be embedded in an X.509 Attribute Certificate (AC) [3] and signed with the policy author's private key.

The Policy Editor is capable of retrieving information from LDAP directories, such as subject names, and writing policy ACs back to the author's entry in the LDAP directory. Authors can use the Policy Editor to browse the directories and select existing policies to update them.

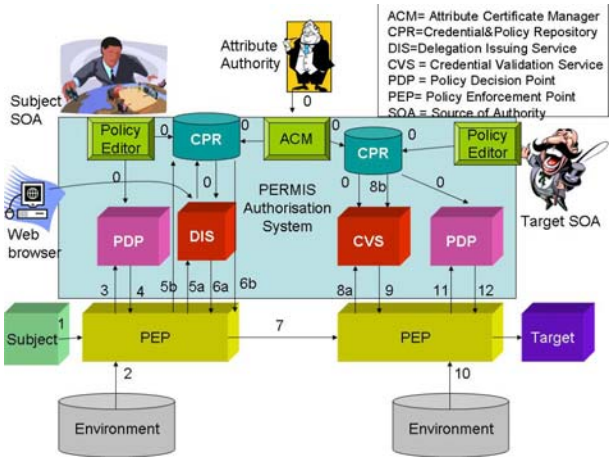


Figure 3: Architecture of the PERMIS Authorization Infrastructure

3.2 Credential Management

The Credential Management system is responsible for issuing and revoking subject credentials. The Attribute Certificate Manager (ACM) tool is used by administrators to allocate attributes to users in the form of X.509 ACs. These bind the issued attributes with the subject's and issuer's identities in a tamper-proof manner. The ACM has a GUI interface that guides the manager through the process of AC creation, modification and revocation. The manager can search for a user in an attached LDAP directory, or enter the DN of the user directly. There is then a picking list of attribute types (e.g. role, affiliation etc.), to which the manager can add his own value (e.g. project manager, University of Kent). There is a pop up calendar allowing the manager to select the dates between which the AC is valid, plus the option of adding appropriate times of day to these. Finally the manager can add a few standard selected extensions to the AC, to say whether the holder is allowed to further delegate or not, and if so, how long the delegation chain can be ("basic attribute constraints" extension [3]), or if the holder may assert the attributes or only delegate them to others ("no assertion" extension [4]). Finally, the manager must add his digital signature to the AC, so the GUI prompts him for the PKCS#12 file holding his private key and his password to unlock it. Once the AC is signed, the manager has the option of storing it in an LDAP directory or local filestore. Besides creating ACs, the ACM allows the manager to edit existing ACs and to revoke existing ACs by deleting them from their storage location. Note that at present revocation lists have not been implemented, because short validity times or deletion from storage have been sufficient to satisfy our current user requirements.

The Delegation Issuing Service is a web service that dynamically issues X.509 ACs on demand. It may be

called directly by an application's PEP after a user has invoked the application, to issue short lived ACs for the duration of the user's task. Alternatively there is a http interface that lets users invoke it via their web browers. This enables users to dynamically delegate their existing longer lived credentials to other users, to enable them to act on their behalf. This is especially powerful, as it empowers users to delegate (a subset of) their privileges to others without any administrative involvement. Because the DIS is controlled by its own PERMIS policy, written by the Subject SOA, an organization can tightly control who is allowed to delegate what to whom, and then leave its subjects to delegate as they see fit. More details of the DIS can be found in [2].

3.3 Authorization Decision Engine

The PERMIS Authorization Decision Engine is responsible for credential validation and access control decision making. Credential validation is the process that enforces the trust model of PERMIS described in Section 2.2. Access control decision making is the process that implements the Role Based Access Control Model described in Section 2.1. The CVS extracts the subset of valid attributes from the set of available credentials, according to the Target SOA's Credential Validation Policy. The PDP makes access control decisions based on the Target SOA's access control policy and the valid attributes passed from the CVS. The PERMIS authorization decision engine is superior to conventional PDPs since it has the ability to validate credentials and delegation chains, which is not a common capability of conventional PDPs e.g. the XACML PDP [15].

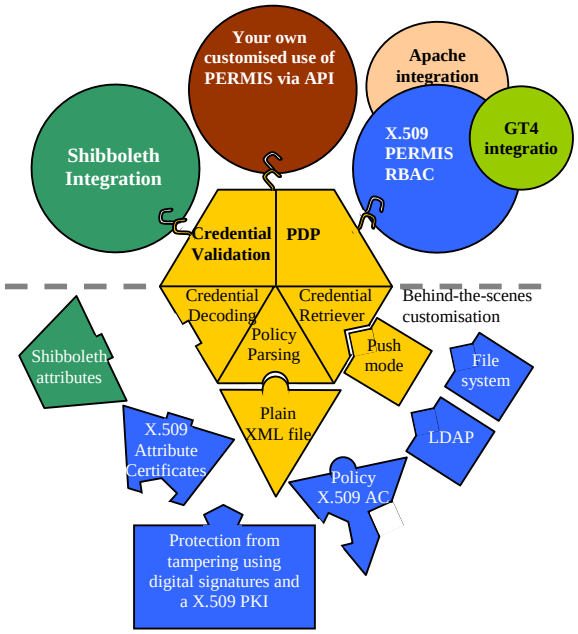


Figure 4: The PERMIS Authorization Decision Engine

As an authorisation infrastructure, PERMIS is not responsible for the actual enforcement of the authorisation decision. This responsibility lies with the application dependent PEP.

Figure 4 depicts the overall architecture of the PERMIS Authorization Decision Engine. It comprises five main components: the PDP, the CVS, the Credential Retriever, the Credential Decoder, and the Policy Parser.

3.4 The PDP

The PDP component is responsible for making access control decisions based on the valid attributes of the user and the Target SOA's access control policy. As stated before, this is based on the Role Based Access Control (RBAC) Model, with support for attribute/role hierarchies.

At initialization time the Target SOA's access control policy is read in and parsed by the Policy Parser so that the PDP is ready to make decisions. Both plain XML policies and digitally signed and protected policies can be read in. The former are stored as files in a local directory whilst the latter are stored as X.509 policy ACs in the LDAP entry of the Target SOA. The latter are tamper resistant and integrity protected, whereas the former have to be protected by the operating system.

Each time the user makes a request to the application to perform a task, the PEP passes this request to the PERMIS PDP along with user's valid attributes and any required environmental attributes such as the time of day. The PEP needs to know which environmental attributes are needed by the access control policy, and since the PEP is software, then it is more likely that the access control policies will be restricted to constraints based on the environmental attributes that the PEP is capable of passing to the PDP.

3.5 The CVS

As described in Section 2.2, all credentials allocated to subjects will be validated by the PERMIS CVS according to the Target SOA's credential validation policy. Figure 5 illustrates the detailed architecture of the CVS, along with the internal data flows.

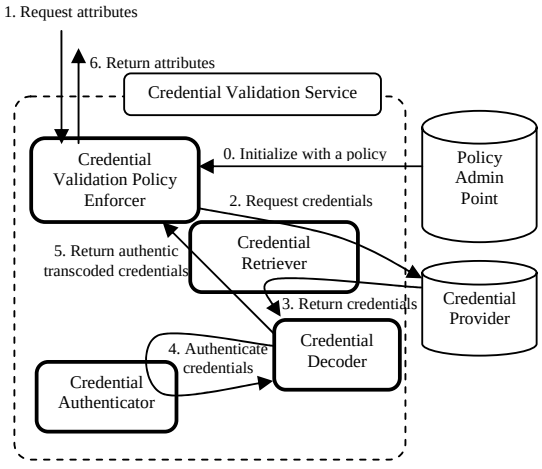


Figure 2 : Data Flow Diagram for Credential Validation Service Architecture

In Figure 5 we can see the general flow of information and sequence of events. First of all the service is initialised by giving it the credential validation policy. The policy parsing module is responsible for this (see Section 3.4). When the user activates the application, the target PEP requests the valid attributes of the subject (step 1). Between the request for attributes and returning them (in step 6) the following events may occur a number of times, as necessary i.e. the CVS is capable of recursively calling itself as it determines the path in a delegation tree from a given credential to a trusted AA specified in the policy.

The Credential Validation Policy Enforcer requests credentials from the Credential Retriever (step 2). PERMIS can operate in either credential pull mode or credential push mode. In credential push mode the application passes the user's credentials along with his request to the target PEP (Step 7 in Figure 3) and the PEP passes them to the CVS (Step 8a in Figure 3). In credential pull mode, the credentials are dynamically pulled from one or more remote credential providers (these could be AA servers, LDAP repositories etc.) by the CVS (step 8b in Figure 3). The actual attribute request protocol (e.g. SAML or LDAP) is handled by the Credential Retriever module. When operating in credential push mode, the PEP stores the already obtained credentials in a local Credential Provider repository and pushes the repository to the CVS, so that the CVS can operate in logically the same way for both push and pull modes. After credential retrieval, the Credential Retriever module passes the credentials to the Credential Decoding module (step 3). From here they undergo the first stage of validation – credential authentication (step 4). Because only the Credential Decoder is aware of the actual format of the credentials, it has to be responsible for authenticating the credentials using an appropriate Credential Authenticator module. Consequently, both the Credential Decoder and Credential Authenticator modules are encoding specific modules. For example, if the credentials are digitally signed X.509 ACs, the Credential Authenticator uses the configured X.509 PKI to validate the signatures. If the credentials are XML signed SAML attribute assertions, then the Credential Authenticator uses the public key in the SAML assertion to validate the signature. The Credential Decoder subsequently discards all unauthentic credentials – these are ones whose digital signatures are invalid. Authentic credentials are decoded and transformed into an implementation specific local format that the Policy Enforcer is able to handle (step 5).

The task of the Policy Enforcer is to decide if each authentic credential is valid (i.e. trusted) or not. It does this by referring to its Credential Validation Policy to see if the credential has been issued by a trusted AA or not. If it has, it is valid. If it has not, the Policy Enforcer has to work its way up the delegation tree from the current credential to its issuer and from there to its issuer, recursively, until a trusted AA is located, or no further issuers can be found (in which case the credential is not trusted and is discarded). Consequently steps 2-5 are recursively repeated until closure is reached (which, in the case of a loop in the credential chain, will be if the same credential is encountered again). Remember that in the general case there are multiple trusted credential issuers, who each may have their own Delegation Policies, which must be enforced by the Policy Enforcer in the same way that it enforces the Target SOA's Delegation Policy.

The CVS can be customized by PERMIS implementers, by either enabling or disabling the credential services built-in with the PERMIS Authorisation Decision Engine, or by implementing their own credential decoding services and plugging them into PERMIS. The latter enables implementers to adopt credential formats that are not implemented by PERMIS, such as local proprietary formats. PERMIS can theoretically be customized to support most application specific credential validation requirements.

4. Integrating PERMIS

4.1 Integration with GT4

Globus Toolkit (GT) is an implementation of Grid software, which has a number of tools that make development and deployment of Grid Services easier [9]. One of the key features of this toolkit is secure communications. However, Globus Toolkit has limited authorisation capabilities based on simple access control lists. To improve its authorization capabilities a Security Assertions Markup Language (SAML) authorization callout has been added. SAML [20] is a standard designed by the Organization for the Advancement of Structured Information Standards (OASIS) to provide a universal mechanism for conveying security related information between the various parts of an access control system. The Global Grid Forum has produced a profile of SAML for use in Grid authorisation [19]. The important consequence of this is that it is now possible to deploy an authorisation service that GT will contact to make authorisation decisions about what methods can be executed by a given subject. A PERMIS Authorisation Service has been developed to provide authorisation decisions for the Globus Toolkit through the SAML callout [8]

4.2 Integration with Shibboleth

Shibboleth [21] is a cross-institutional authentication and authorisation architecture for single sign on and access control of web resources. Shibboleth defines a protocol for carrying authentication information and user attributes from the user's home site to the resource site. The resource site can then use the user attributes to make access control decision about the user's request. A user only needs to be authenticated once by the home site in order to visit other Shibboleth protected resource sites in the federation, as the resulting authentication token is recognized by any member of the federation. In addition to this, protection of the user's privacy can be achieved, since the user is able to restrict what attributes will be released to the resource providers from his/her home site. However Shibboleth's built in access control decision making based on the user's attributes, is simplistic in its functionality, and the

management of the access controls is performed together with web server administration at the resource site. Furthermore, distributed management of credentials and dynamic delegation of authority are not supported. To rectify these deficiencies, a Shibboleth-Apache Authorisation Module (SAAM) has been developed which integrates PERMIS with Shibboleth. SAAM plugs into Apache and replaces the Shibboleth authorization functionality with calls to the PERMIS authorization decision engine. A full description is provided in [5]

PERMIS extends the access control model used in Shibboleth by introducing hierarchies of roles, distributed management of attributes, and policy controlled decisions based on dynamically evaluated conditions. PERMIS supports the existing semantics of Shibboleth attributes, but also allows X.509 ACs to be used instead, where more secure credentials are needed.

4.3 Integration with GridShib

GridShib [10] provides interoperability between Globus Toolkit [9] and Shibboleth [21]. The GridShib Policy Information Point (PIP) retrieves a user's attributes from the Shibboleth Identity Provider (IdP). These attributes are parsed and passed to the GT4 PEP. The GT4 PEP then feeds these attributes to the corresponding PDP to request an authorisation decision. GridShib integrates Shibboleth's attribute management functionality with GT4's authorisation decision making for Grid jobs. However, like GT4, GridShib provides only limited PDP functionality, which is based on access control lists and is not capable of coping with dynamically changing conditions, which a policy based engine is.

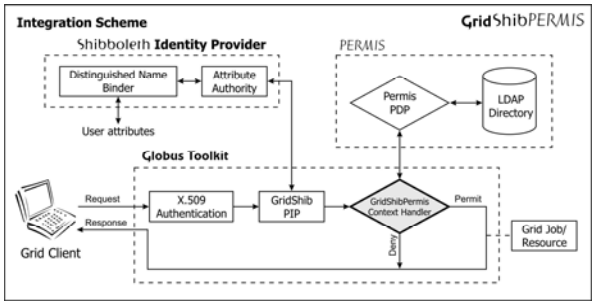


Figure 6: GridShibPERMIS Integration Scheme

Figure 6 shows the GridShibPERMIS integration scheme. GridShibPERMIS provides a GridShibPERMIS Context Handler that can be integrated with GT4 as a callable PDP. The Context Handler is invoked by GT4 when an authorisation decision is to be made. The Context Handler is fed with the user's attributes that have been retrieved from the Shibboleth IdP. They are parsed and stored in a local Credential Provider Repository, ready to be accessed by the PERMIS CVS as described in Section 3.5. The Context Handler calls the PERMIS CVS followed by the PDP, which renders an access control decision based on the Target SOA's policy, and returns it to GT4. In the case of multiple PDPs being configured into GT4, the final authorisation decision is made based on combining all the decisions returned by all the different PDPs. The combining algorithm currently adopted by GT4 is "Deny-Override", which means that the user's request is authorised if and only if no PDP denies the request and at least one PDP grants it.

5. Related Work

Manandhar et at. [12] present an application infrastructure in which a data portal allows users to discover and access data over Grid systems. They propose an authorization framework that allows the data portal to act as a proxy and exercise the user's privileges. When a user authenticates to the data portal, a credential is generated stating that the data portal is authorized to exercise the user's privileges for a specific period. The credential is then used by the data portal to retrieve the user's authorization tokens from various providers. When requesting a service from a service provider, the data portal presents both the credential and the authorization tokens. The authorization decision is then made by the service provider. The proposed infrastructure mainly focuses on the interaction between different systems in the Grid environment, with no in depth discussion about the access control model or the trust model. Credential verification is also missing from the discussion.

XACML [14] defines a standard for expressing access control policies, authorization requests, and authorization responses in XML format. The policy language allows users to define application specific data types, functions, and combining logic algorithms, for the purpose of constructing complex policies. Sun's open source XACML implementation [15] is a java implementation of the XACML 2.0 standard and provides most of the feature in the standard. The XACML policy language is richer than that of PERMIS's PDP policy, but XACML has not yet addressed the issue of credential validation and is only now working on dynamic delegation of authority [22].

The Community Authorisation Service (CAS) [11] was developed by the Globus team to improve upon the manageability of user authorisation. CAS allows a resource owner to grant access to a portion of his/her resource to a VO (or community – hence the name CAS), and then let the community determine who can use this allocation. The resource owner thus partially delegates the allocation of authorisation rights to the community. This is achieved by having a CAS server, which acts as a trusted intermediary between VO users and resources. Users first contact the CAS asking for permission to use a Grid resource. The CAS consults its policy (which specifies who has permission to do what on which resources) and if granted, returns a digitally self-signed capability to the user optionally containing policy details about what the user is allowed to do (as an opaque string). The user then contacts the resource and presents this capability. The resource checks that the capability is signed by a known and trusted CAS and if so maps the CAS's distinguished name into a local user account name via the Gridmap file. Consequently

the Gridmap file now only needs to contain the name of the trusted CAS servers and not all the VO users. This substantially reduces the work of the resource administrator. Further, determining who should be granted capabilities by the CAS server is the task of other managers in the VO community, so this again relieves the burden of resource managers. For finer grained access control, the resource can additionally call a further routine, passing to it the opaque policy string from the capability, and using the returned value to refine the access rights of the user. Unfortunately this part of the CAS implementation (policy definition and evaluation routine) were never fully explored and developed by the Globus team. This is precisely the functionality that PERMIS has addressed.

The main purpose of SPKI [16] is to provide public key infrastructures based on digital certificates without depending upon global naming authorities. SPKI binds local names and authorizations to public keys (or the hash values of public keys). Names are allocated locally by certificate issuers, and are only of meaning to them. SPKI allows authorizations to be bound directly to public keys, removing the process of mapping from authorization to names and then to public keys. SPKI supports dynamic delegation of authorizations between key holders, and allocation of authorizations to groups. Though SPKI can convey authorization information, it does not cover authorization decision making or access control policy issues. One can thus regards SPKI as an alternative format to X.509 ACs or SAML attribute assertions for carrying credentials, and PERMIS could easily be enhanced to support this format of credential if it were required.

The EU DataGrid and DataTAG projects have developed the Virtual Organisation Membership Service (VOMS) [6] as a way of delegating the authorisation of users to managers in the VO. VOMS is a credential push system in which the VOMS server digitally signs a short lived X.509 role AC for the VO user to present to the resource. The AC contains role and group membership details, and the Local Centre Authorisation Service (LCAS) [7] makes its authorisation decision based upon the user's AC and the job specification, which is written in job description language (JDL) format. This design is similar in concept to the CAS, but differs in message format and syntax. However what neither VOMS nor CAS nor LCAS provide is the ability for the resource administrator to set the policy for access to his/her resource and then let the authorisation infrastructure enforce this policy on his/her behalf. This is what systems such as PERMIS and Keynote [17] provide. It would therefore be relatively easy to replace LCAS with the PERMIS decision engine, so that VOMS allocates role ACs and pushes them to the resource site, whilst PERMIS makes the policy controlled authorization decisions.

KeyNote [17] is a trust management system that provides a general-purpose mechanism for defining security policies and credentials, and rendering authorization decisions based on security policies and credentials. KeyNote provides a language for defining both policies and assertions, where policies state the rules for security control, and assertions contain predicates that specify the granted privileges of users. KeyNote has been implemented and released as an open source toolkit. But KeyNote is not without its limitations. Keynotes policies and credentials are in their own proprietary format. KeyNote credentials have no time limit, and Keynote has no concept of revocation of credentials. Further, policies define the roots of trust, but the policies themselves are not signed and therefore have to be stored securely and are only locally trusted.

6. Conclusions

This paper presents our work on building a modular authorization infrastructure, PERMIS. We have explained the conceptual models of PERMIS by describing the authorization model, the access control model, and the trust model of PERMIS. The design and the implementation of PERMIS have also been presented, with details of the architecture and an explanation of the facilities PERMIS provides to support policy management, attribute management, and decision making. Details of the decision making process and the credential validation service are also given, showing how PERMIS implements hierarchical RBAC decision making based on user credentials and various authorization policies.

Finally, we have presented a comparison of related work, pointing out their relative advantages and disadvantages as compared to PERMIS.

6.1 Future Work

In an application, sometimes coordination is needed between access control decisions. For example, in order to support mutually exclusive tasks (Separation of Duties), the PDP needs to know if the same user is trying to perform a second task in a set of mutually exclusive ones. Alternatively, if multiple resources are available but their use is to be restricted, for example a maximum of 30GB of storage throughout a grid, then enforcing this becomes more difficult. The use of a stateful PDP allows coordination between successive access control decisions, whilst passing coordination data between PDPs allows coordination over the use of restricted multiple resources. In order to achieve the latter, the access control policies should state what coordination between decision making is needed, which coordination data is used to facilitate this, and how this coordination data is updated afterwards. We have already implemented Separation of Duties in a prototype stateful PDP and we are currently working on more sophisticated distributed coordination between PDPs.

Obligations are actions that are required to be fulfilled on the enforcement of access control decisions. Existing authorization infrastructures are mainly concerned with access control decision making, but this is not sufficient in scenarios where obligations are needed. XACML policies already support obligations and we are currently incorporating these into PERMIS. We are building an obligation engine that will evaluate the obligations that are embedded in a policy e.g. Add the amount requested to the

amount already consumed, and will return the obligated action to the PEP for enforcement, since it is the PEP that ultimately has to interpret and fulfill the obligations.

Acknowlegements

We would like to thank the UK JISC for funding part of this work under the DyCOM, DyVOSE, SIPS and GridAPI projects, and the EC for funding part of this work under the TrustCoM project (FP6 project number 001945).

References

[1] D.W.Chadwick, A. Otenko "The PERMIS X.509 Role Based Privilege Management Infrastructure". Future Generation Computer Systems, 936 (2002) 1–13, December 2002. Elsevier Science BV.

[2] D.W.Chadwick. "Delegation Issuing Service". NIST 4th Annual PKI Workshop, Gaithersberg, USA, April 19-21 2005

[3] ISO 9594-8/ITU-T Rec. X.509 (2001) "The Directory: Public-key and attribute certificate frameworks"

[4] ISO 9594-8/ITU-T Rec. X.509 (2005) "The Directory: Public-key and attribute certificate frameworks"

[5] Wensheng Xu, David Chadwick, Sassa Otenko. "Development of a Flexible PERMIS Authorisation Module for Shibboleth and Apache Server". Proceedings of 2nd EuroPKI Workshop, University of Kent, July 2005

[6] R. Alfieri et al. "VOMS: an Authorization System for Virtual Organizations", 1st European Across Grids Conference, Santiago de Compostela, February 13-14, 2003

[7] Martijn Steenbakkers "Guide to LCAS v.1.1.16", Sept 2003. Available from http://www.dutchgrid.nl/DataGrid/wp4/lcas/edg-lcas-1.1

[8] David Chadwick, Sassa Otenko, and Von Welch. "Using SAML to Link the GLOBUS Toolkit to the PERMIS Authorisation Infrastructure". In Proceedings of Eighth Annual IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, Windermere, UK, September 2004.

[9] I. Foster. "Globus Toolkit Version 4: Software for Service-Oriented Systems". IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2005.

[10] Barton, T., Basney, J., Freeman, T., Scavo, T., Siebenlist, F., Welch, V., Ananthakrishnan, R., Baker, B., and Keahey, K. "Identity Federation and Attribute-based Authorization through the Globus Toolkit, Shibboleth, Gridshib, and MyProxy", 5th Annual PKI R&D Workshop. April 2006.

[11] Ian Foster, Carl Kesselman, Laura Pearlman, Steven Tuecke, and Von Welch. "The Community Authorization Service: Status and Future". In Proceedings of Computing in High Energy Physics 03 (CHEP '03), 2003.

[12] Ananta Manandhar, Glen Drinkwater, Richard Tyer, Kerstin Kleese. "GRID Authorization Framework for CCLRC Data Portal", Second Earth Science Portal Workshop: Web Portal Framework Design/Implementation, September 2003.

[13] Sacha Brostoff, M. Angela Sasse, David Chadwick, James Cunningham, Uche Mbanaso, Sassa Otenko. "*"R-What?"* Development of a Role-Based Access Control (RBAC) Policy-Writing Tool for e-Scientists" Software: Practice and Experience
Volume 35, Issue 9, Date: 25 July 2005, Pages: 835-856

[14] OASIS. "XACML 2.0 Core: eXtensible Access Control Markup Language (XACML) Version 2.0", Oct, 2005.

[15] Sun's XACML Implementation available on http://sunxacml.sourceforge.net/.

[16] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomsa, and T. Ylonen. "SPKI Certificate Theory". RFC 2693, September 1999.

[17] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. "The KeyNote Trust Management System Version 2". RFC 2704, Sept. 1999.

[18] David F. Ferraiolo and Ravi Sandhu and Serban Gavrila and D. Richard Kuhn and Ramaswamy Chandramouli. "Proposed NIST standard for role-based access control". ACM Transactions on Information and System Security Volume 4, Issue 3. August 2001.

[19] Von Welch, Rachana Ananthakrishnan, Frank Siebenlist, David Chadwick, Sam Meder, Laura Pearlman. "Use of SAML for OGSI Authorization", Aug 2005, Available from https://forge.gridforum.org/projects/ogsa-authz

[20] OASIS. "Security Assertion Markup Language (SAML) 2.0 Specification", November 2004.

[21] S. Cantor. "Shibboleth Architecture, Protocols and Profiles", Working Draft 02. 22 September 2004, see http://shibboleth.internet2.edu/

[22] XACML v3.0 administration policy Working Draft 05 December 2005. http://www.oasis-open.org/committees/documents.php?wg abbrev=xacml.

[23] N. Zhang, L. Yao, A. Nenadic, J. Chin, C. Goble, A. Rector, D. Chadwick, S. Otenko and Q. Shi; "Achieving Fine-grained Access Control in Virtual Organisations", to appear in Concurrency and Computation: Practice and Experience, published by John Wiley and Sons publisher.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

A View Based Security Framework for XML

Wenfei Fan* Irini Fundulaki Floris Geerts† **Xibei Jia** Anastasios Kementsietsidis
University of Edinburgh
{wenfei@inf,efountou@inf,fgeerts@inf,x.jia@sms,akements@inf}.ed.ac.uk

Abstract

As science communities and commercial organisations increasingly exploit XML *as a means of exchanging and disseminating information, the selective exposure of information in* XML *has become an important issue. In this paper, we present a novel* XML *security framework developed to enforce a generic, flexible access-control mechanism for* XML *data management, which supports efficient and secure query access, without revealing sensitive information to unauthorized users. Major components underlying the framework are discussed and the current status of a reference implemenation is reported.*

1 Introduction

The wide adoption of XML related technologies in UK eScience projects clearly demonstrates the importance of XML data management. With the prevalent use of XML in science communities and commercial organizations, the selective exposure of information in XML to safeguard data confidentiality, privacy and intellectual property has become a primary concern for data providers, curators and consumers. As a result, there is a growing need for a generic, flexible access control framework for XML data that supports efficient and secure query access, without revealing sensitive information to unauthorized users.

More specifically, for an XML database there might be multiple user groups who wish to query the same XML document. For these user groups different *access policies*[1] may be imposed, specifying the portions of the document the users are granted or denied access to. Figure 1 shows a simplified example of a medical records XML database. As shown in the figure, the security administrator could see the whole database; a doctor can only access the records of his patients; a patient can access his own medical records; an insurer can only read his customers' billing information; and a medical researcher could retrieve the diagnosis data for research purposes, but not the information on doctors or patients.

These *security constraints* are specified in the access policies. Note that not only the *data* (i.e., actual values) but also the *structural information* could be protected. The goal of a security system is to ensure that the evaluation of a user query over the XML database returns only information in the database that the user is allowed to access; in other words, we seek to protect sensitive data from direct access or indirect inference through queries by unauthorized users.

Addressing such security concerns mandates the development of a generic, flexible access control framework that provides:

1. a rich *security specification language* that supports the definition of access policies at various granularity levels (e.g., restricting access to entirely subtrees or specific elements in the document tree based on their content or location).
2. *schema availability*: XML documents are typically accompanied by a DTD or an XML Schema that specifies the internal structure of the data. For all the reasons that a database schema is needed for query formulation and processing in traditional databases, schemas are also important for XML query formulation, optimization, XML data exchange and integration. In the context of access control, the availability of an XML schema that specifies the structure of accessible data is critical to the users who can then formulate queries only over this data.
3. *efficient enforcement* of security constraints during XML query evaluation. The enforcement of such constraints should not imply any drastic degradation in either performance or functionality for the underlying XML query evaluation engine.

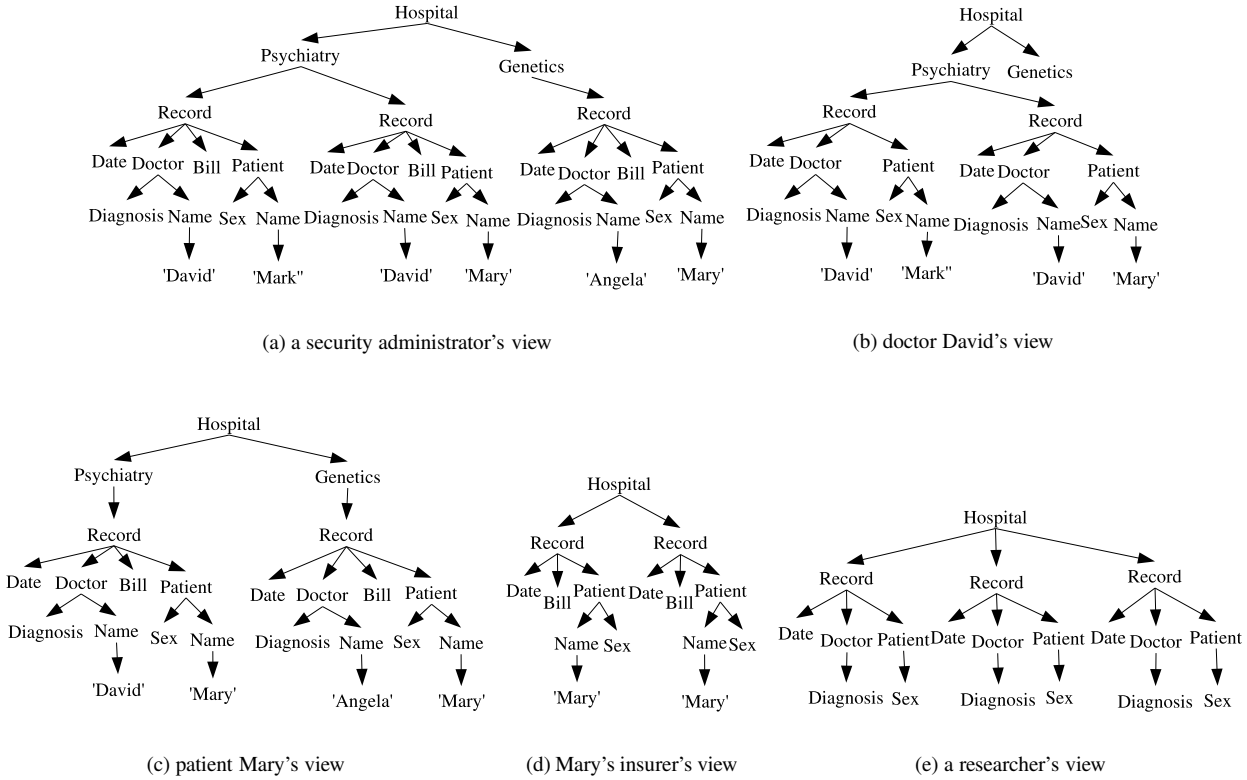
In this paper, we propose an XML security framework and explain the components in it. The security framework supports:

- a powerful *specification language* for the definition of rich access policies;

*Supported in part by EPSRC GR/S63205/01, GR/T27433/01 and BBSRC BB/D006473/1. Wenfei Fan is also affiliated to Bell Laboratories, Murray Hill, USA.

† Floris Geerts is a postdoctoral researcher of the FWO Vlaanderen and is supported in part by EPSRC GR/S63205/01. He is also affiliated to Hasselt University and Transnational University of Limburg, Belgium.

[1] From this point on and w.l.g. we use intercheangably the terms security specification and access policy.



(a) a security administrator's view

(b) doctor David's view

(c) patient Mary's view

(d) Mary's insurer's view

(e) a researcher's view

Figure 1. Different access privileges on a medical records XML database

- the ability to effectively derive and publish a number of different *schemas* (DTD or XML Schema) characterizing accessible data based on different access policies;
- novel *query processing and optimization techniques* that enforce the security constraints specified in access policies by ensuring efficient and secure query access to large XML documents for a sufficiently rich and powerful query language.

Related Work: Although a number of XML security models have been proposed [1, 2, 3, 4, 9, 14, 15, 16], these models may either reject proper queries and access [9, 16], incur costly run-time security checks [15], require expensive view materialization and maintenance [3, 4], or complicate integrity maintenance by annotating the underlying data [2]. Perhaps a more subtle problem is that none of these earlier models provides users with a schema characterizing the information they are allowed to access. Worse still, some models expose the full document schema to all users, and make it possible to employ (seemingly secure) queries to infer information that the access policy was meant to protect.

The architecture, algorithms and reference implementation presented in this paper form the first security framework that supports flexible access policy specifications, access and inference control, efficient enforcement techniques for access policies, and schema availability.

2 The Security Architecture

Views are commonly used in traditional (relational) databases to enforce access control, and therefore provide a promising method to access XML data securely. Past experience has shown that these views should be *virtual*. The reason is evident: a large number of user groups may want to query the same XML document, each with a different access policy. To enforce these policies, we may provide each user group with an XML view consisting of only the information that the users are allowed to access, such that users may query the underlying data only through their view. The views should be kept virtual since it is prohibitively expensive to materialize and maintain a large number of views, one for each user group. Therefore, a query posed by users on a *virtual view* will be rewritten to an equivalent one that will be evaluated on the *underlying* XML document.

The architecture of the security framework we propose is based on the novel concept of *security views* [5], that provide for each user group a virtual XML view consisting of all and only the information that the users are authorized to access, and a view schema that this XML view conforms to.

As depicted in Figure 2, the architecture consists of four core modules (*view derivation* and *query rewriting*, *evaluation* and *optimization*), three secondary modules (*security specification editor*, *query editor*, *result viewer*) and an optional *indexer* module. Briefly, the secondary modules implement a user-friendly interface through which the user

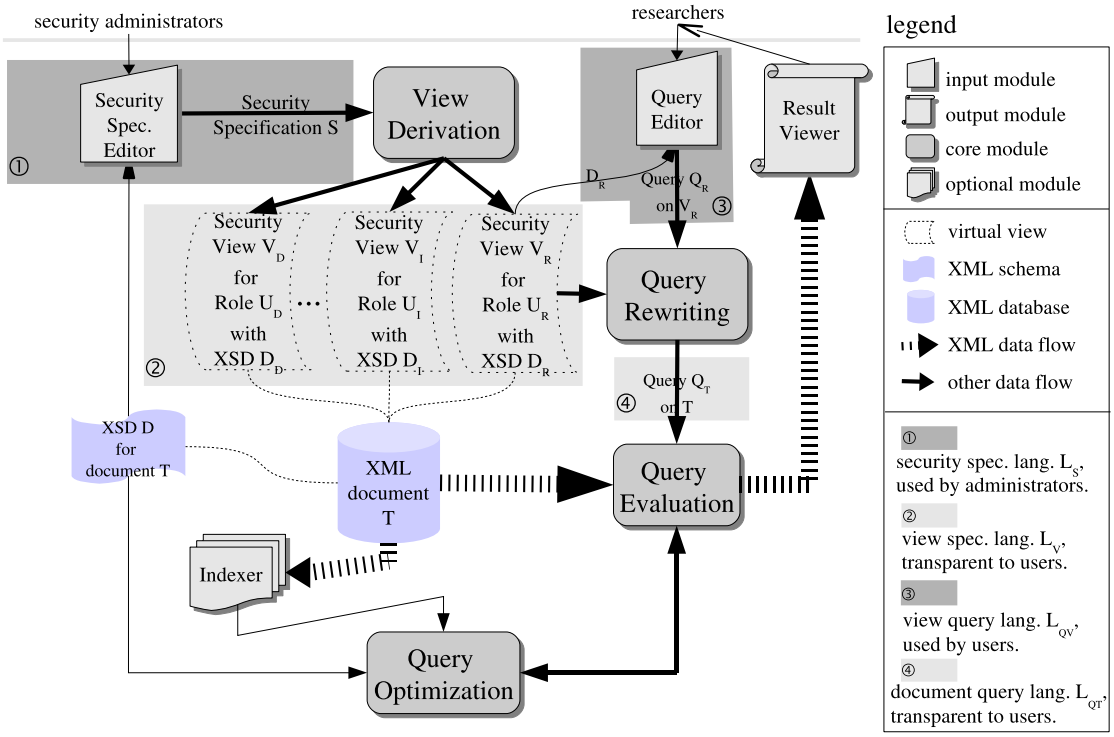


Figure 2. The proposed security architecture

interacts with the system. These modules provide forms for user input and are responsible for presenting the system output (e.g. the result of user queries). The core and indexer modules implement all the basic algorithms and functionality of the system. These latter modules are internal and thus users do not directly interact with them.

There are two types of users that interact with the system, namely, *security administrators* who specify the access policies, and *normal users in certain roles* who query the XML views to which they have been granted access through their roles.

Central to the architecture is a number of *security views* which is automatically derived from the access policies. These policies are manually specified by annotating the schema of the XML documents.

Four languages are involved in the architecture: *security specification* and *view specification* languages (L_S and L_V resp.) and *view query* and *document query* languages (L_{QV} and L_{QT} resp.). Among them, the security specification language L_S and view query language L_{QV} are used by the security administrators and the users respectively. The view specification language L_V and document query language L_{QT} are used by the view derivation and query rewriting modules respectively, for representing the automatically generated view definitions and queries. Unlike the former two, they are internal and thus not visible to users. We have to note here, that in contrast to most of the state-of-the-art approaches for XML access control that employ XPath as their security specification language [8], we use Regular XPath, a mild extension of XPath. We argue in Section 4 why we choose Regular XPath as the security view specification language over the normal XPath.

We will use the example introduced in Section 1 to illustrate the security framework built on this architecture. First, security administrators specify the security specifications S for the different roles (researchers, insurers, doctors) using the *security specification module*. The security specifications S are then passed to the *view derivation module* where a set of security view definitions V_R, V_I and V_D are automatically derived from S for the roles of researchers, insurers, and doctors respectively. Next, a user U_R in the role of, say, a researcher, poses a query Q_R using the *query editor module* over the virtual security view V_R. This query Q_R is subsequently rewritten into a new query Q_T over the underlying document T in the *query rewriting module*. Query Q_T is optimised in the *query optimization module* with the optional index from the *indexer module*, and passed to the *query evaluation module* in order to be executed over the document T. Finally, the results of Q_T (and therefore Q_R) are shown in the *result viewer module* to the user U_R.

The eight modules that comprise the architecture can also be classified in two categories according to their purpose: modules to enforce security constraints (security specification editor, view derivation); modules to support efficient query answering (query editor, query rewriting, evaluation and optimization, indexer, result viewer). In the following sections, we use this view of modules to present, in more detail, their specifics.

3 Modules to Enforce Security Constraints

To enforce the security constraints, a security framework needs the necessary tools to *(i)* support the specification of these constraints and *(ii)* enforce them when the users interact with the system. In the proposed framework, these functions are modularized into the *security specification editor* and *view derivation* module.

3.1 Security Specification Editor

• **Input:** XML schema (XSD) D of the underlying document T.

• **User Interaction:** the security administrators associate the security constraints for each role R with the elements in XSD D.

• **Output:** security specification S.

The first step in the security framework is the definition of access policies by the security administrators. Multiple such policies are possibly declared over T at the same time, each specifying for a class of users in a certain role the elements in T the users are granted, denied, or conditionally granted access to.

In contrast to relational databases for which access control can be specified via a view defined with an SQL query, access policies for hierarchical XML data must be specified at various levels of granularity based on both the *structure* and the *values* of the data.

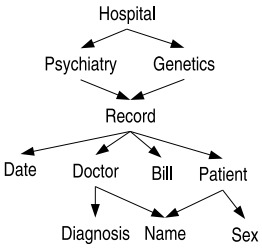


Figure 3. The XSD graph of the medical records XML database

We have defined a fine-grained security specification language L_S and implemented it in the security specification editor. A *security specification* S expressed in the language L_S is a simple extension of the XML schema document (XSD) D associating element types with security annotations (Regular XPath qualifiers). These security annotations specify structure- and content-based accessibility of the XML elements in T, instances of the element types in D. Specifically, S extends D as follows: it allows the definition of a "role" element with an "accessible" attribute inside each "element" type E in D to specify the security constraint for type E. The text content of this "role" element is a role name indicating which role is concerned by S. The value of the "accessible" attribute could be "yes", "no" or a Regular XPath qualifier enclosed in "[.]", which indicates that the elements of type E in an instantiation of D are accessible, inaccessible, and conditionally accessible, respectively. The value of the attribute "accessible" for the root of D is "yes" for every role by default, unless otherwise specified. A fragment of the security specification S for the XML database in Figure 1 is illustrated in the following example. It is defined by extending the XSD graph shown in Figure 3.

```
<xs:element name="Hospital">
  <xs:complexType><xs:sequence>
    <xs:element name="Psychiatry" type="records">
      <as:role accessible="no">insurer researcher</as:role>
    </xs:element>
    <xs:element name="Genetics" type="records">
      <as:role accessible="no">insurer researcher</as:role>
    </xs:element>
  </xs:sequence></xs:complexType>
</xs:element>
<xs:complexType name="records"><xs:sequence>
  <xs:element name="Record" maxOccurs="unbounded">
    <as:role accessible="[Patient/Name=$custName]">insurer</as:role>
    <as:role accessible="[Doctor/Name=$me]">doctor</as:role>
    <as:role accessible="yes">researcher</as:role>
    <xs:complexType><xs:sequence>
      <xs:element name="Date" type="xs:date"/>
      <xs:element name="Doctor" type="dType">
        <as:role accessible="no">insurer</as:role>
      </xs:element>
      <xs:element name="Bill" type="bType">
        <as:role accessible="no">doctor researcher</as:role>
      </xs:element>
      <xs:element name="Patient" type="pType"/>
    </xs:sequence></xs:complexType>
  </xs:element>
</xs:sequence> </xs:complexType>
<xs:complexType name="dType"><xs:sequence>
  <xs:element name="Diagnosis" type="diaType"/>
  <xs:element name="Name" type="xs:string">
    <as:role accessible="no">researcher</as:role>
  </xs:element>
</xs:sequence></xs:complexType>
<xs:complexType name="pType"><xs:sequence>
  <xs:element name="Sex" type="xs:sexType"/>
  <xs:element name="Name" type="xs:string">
    <as:role accessible="no">researcher</as:role>
  </xs:element>
</xs:sequence></xs:complexType>
...
```

Figure 4. Security specification S

Example 1: We give in Figure 4 the security specification S over the XSD D shown in Figure 3. The annotations *as:role* over schema D for the different roles U_i form the security specification, which classifies the nodes in the XML document into three sets: (a) *accessible nodes*, *e.g.,* the root *Hospital* for role insurer; (b) *inaccessible nodes*, *e.g.,* the *Psychiatry* and *Genetics* elements for role insurer or researcher;

and (c) *conditional accessible nodes*, *e.g., Record* elements, which are accessible by a doctor if and only if the record is about one of his patients.

The security specification language L_S supports the following salient features: (a) *inheritance*: if a node does not have an accessibility explicitly defined, then it inherits the accessibility of its parent (the root node is accessible by default for all roles); (b) *overriding*: on the other hand, an explicit accessibility definition at a node will override the accessibility inherited from its parent; (c) *content-based access privileges*: the accessibility of a node is specified with predicates/qualifiers (yes, no, or Regular XPath qualifiers); and (d) *context-dependency*: the accessibility of elements in a document is determined by paths from the root to these elements in the document; *e.g.,*, the *diagnosis* child of a *doctor* is accessible only if the *doctor* is accessible.

The security specification editor provides the security administrators two modes to fill in the access policy: a graph mode and a text mode. The graph mode presents the administrators an XSD graph of the underlying XML document. The administrators associate a security annotation for role U_i on node E_j by clicking a node E_j in the XSD graph. The text mode automatically generates code skeletons in the access specification language L_S. The administrators can add more rules to override the default rules in the skeletons to (conditionally) grant access to any part of the XML document for some role

3.2 View Derivation Module

• **Input:** security specifications S.

• **Output:** *i)* security view specifications V_i for each role U_i, *ii)* security view schema D_i for each role U_i.

The security specification S is subsequently processed by the view derivation module, which automatically derives a set of *security views* V_1, V_2, ..., V_n expressed in an internally used view specification language L_V. A view V is an XSD D_v annotated with Regular XPath queries along the same lines as DAD (IBM DB2 XML Extender [10]), AXSD (Microsoft SQL Server [13]) and Oracle [17]). The XSD D_v exposes only accessible data *w.r.t.* S, and is used by the users authorized by S to formulate their queries *over the view*. The Regular XPath annotations are not visible to authorized users, and are used internally by the system to extract accessible data from the XML document T. The only structural information about T that the users are aware of is D_v, and no information beyond the view can be inferred from user queries. Thus, our security views support both access/inference control and schema availability.

Syntactically, a security view V_i from S to an XSD D_i for role U_i, denoted by $V_i : S \rightarrow D_i$ is defined as a pair $V_i = (D_i, \sigma_{ij})$, where σ_{ij} defines Regular XPath query annotations used to extract accessible data for each element type E_j in D_i from an instance T of D. Specifically, for each *element* E_j in D_i, σ_{ij} is a *data* element which contains a Regular XPath query defined over instances of D. If not explicitly defined, the default σ_{ij} for an element E_j is simply a trivial relative Regular XPath query that is the same as the element name.

An access control policy over a relational database can be simply enforced by defining a view via an SQL query. The view, referred to as a *security view*, consists of all and only the accessible information *w.r.t.* the policy. The view schema – flat relations – can be readily derived from the view definition and provided to the users for their query formulation. In contrast, security views for XML pose much new challenges. Consequently, complex algorithms are needed for the view derivation module. A dynamic programming based view derivation algorithm has been proposed in [5], which assumes DTDs and uses a subset of XPath to define specifications and security views. These algorithms which consider XPath are already extended in our context to account for Regular XPath.

```
<xs:element name="Hospital">
  <xs:complexType><xs:sequence>
    <xs:element name="Record" maxOccurs="unbounded">
      <vs:data>
      (Psychiatry | Genetics)/Record[Patient/Name=$custName]
      </vs:data>
        <xs:complexType><xs:sequence>
          <xs:element name="Date" type="xs:date"/>
          <xs:element name="Bill" type="bType"/>
          <xs:element name="Patient" type="pType"/>
        </xs:sequence></xs:complexType>
      </xs:element>
    </xs:sequence></xs:complexType>
  </xs:element>
  ...
```

Figure 5. Security view V_I for role "insurer"

Example 2: Figure 5 shows the security view derived for the role "insurer" from the access specification S in Example 1.

4 Modules to Support Efficient Query Answering

A security framework should not only be able to enforce but also to support efficient querying in the presence of security constraints. The latter goal is important because a security framework with a dramatically performance decrease for query answering cannot be useful to real world users and applications.

Since security views are virtual, user queries must be rewritten into queries on the underlying XML document and then executed on the XML database to generate the result. In what follows, we present (through the modules) the whole query answering process starting from the moment queries are composed in the query editor all the way to the point where query results are returned to the user.

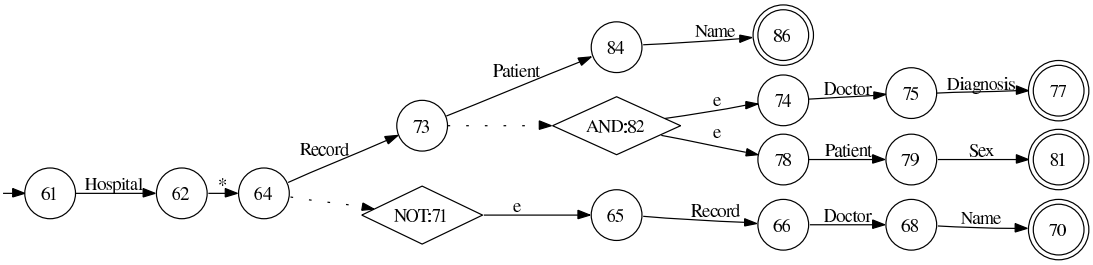


Figure 6. The MFA $\mathcal{M}_0$ characterizing query Q_0

4.1 Query Editor Module

• **Input:** XSD D_i of the security view V_i.

• **User Interaction:** formulation of queries over the XSD D_i by user in role U_i.

• **Output:** query Q_i over the view V_i.

After the security views have been either derived by the view derivation module, or manually specified by the security administrators, the system is ready to answer user queries. These are composed in the *query editor module*. The module provides the users in role U_i their corresponding view XSD D_i to guide them during query composition. This is an important feature for the users because without such a schema the query formulation process is easily fallen into repeated failures: the users continually query some information they are not allowed to access as they do not have a clear picture of the accessible information. In addition, equipped with the security view schema, it becomes possible for the query editor module to provide more assistance to users such as auto-completion (offer hints to the users about next possible tokens in the query being composed according to the query grammar and the view schema) or visual query composition (allow users to compose queries graphically through drag and drop operations).

4.2 Query Rewriting Module

• **Input:** *(a)* security view specification V_i for role U_i and *(b)* query Q_i posed over V_i.

• **Output:** query Q_T over the XML document T.

Once query Q_i on view V_i is passed to the query rewriting module, it is rewritten to a new query Q_T on the underlying document. Nevertheless, this query rewriting is non-trivial. For example, XPath, the core of XQuery and XSLT, is *not closed under rewriting*, *i.e.,* for an XPath query on a recursively defined view there may not exist an equivalent XPath query on the underlying document [6]. Therefore, an extension is needed for XPath to serve as a view specification language L_V (which is closed under rewriting). In our security framework, we adopt Regular XPath along with XSD as the security view specification language. Here Regular XPath is a mild extension of XPath that supports the general Kleene closure $(.)^*$ instead of the limited recursion '//' (descendant-or-self axis). Therefore, user queries already written in XPath can be used *as-is* and need not be re-defined, a necessity if a richer language like XQuery or XSLT was used. Second, and more to the point, Regular XPath is closed under rewriting for XML views, recursively defined or not. Since Regular XPath subsumes XPath, any XPath query posed on any XML view can be rewritten to an equivalent Regular XPath query on the underlying data. This also justifies the decision of using Regular XPath as the security annotations in the security specification language. In contrast to other frameworks supporting the specification of XML views, this design choice makes the implemented system capable of handling recursively defined schema (and thus views).

Apart from the closure of the query language, another important concern is the size of the rewritten query. Indeed, [6] has shown that the size of the rewritten query Q_T, if directly represented in Regular XPath, may be exponential in the size of input query Q_i. The query rewriting model overcomes this challenge by employing an automaton characterization of Q_T, denoted by MFA (*mixed finite state automata*), which is *linear* in the size of Q_i. An MFA of Q_T is a finite state automaton (NFA, characterizing the data-selection path of Q_T) annotated with alternating automata (AFA, capturing the predicates of Q_T).

Example 3: Figure 6 depicts the MFA $\mathcal{M}_0$ characterizing the Regular XPath query:

Q_0 = Hospital/*[*not*(Record/Doctor/Name='David')]/
Record[Doctor/Diagnosis='Haemophilia'
and Patient/Sex='Female']/Patient/Name

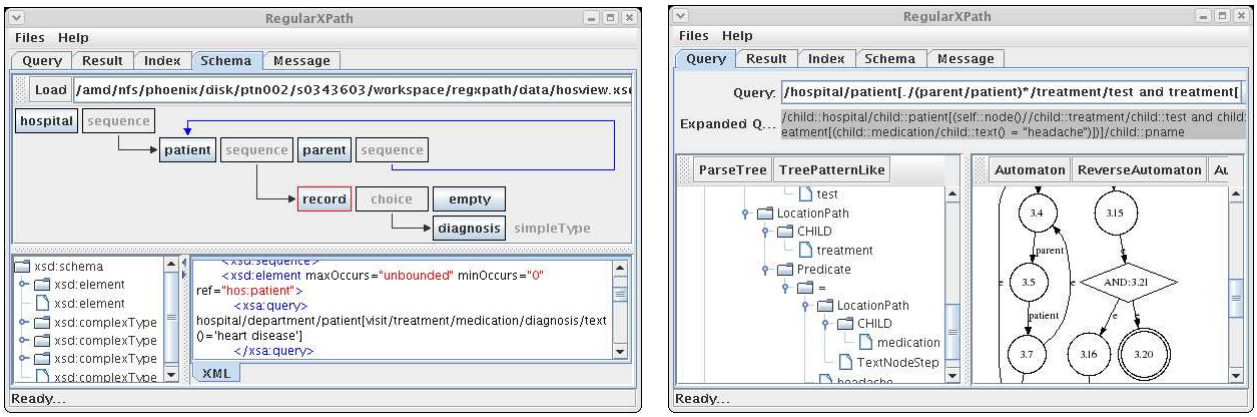
In the MFA $\mathcal{M}_0$, the NFA consists of states $(61, 62, 64, 73, 84, 86)$ and represents the selection path *Hospital/*/Record/Patient/Name*; it is annotated with an AFA (linked to states 64 and 73 via dotted arrows) capturing the predicates of Q_0 (the part enclosed in [ ]).

4.3 Query Evaluation and Optimization Modules

• **Input:** *(a)* XML document T and *(b)* query Q_T over T.

• **Output:** the answer of query Q_T.

After query Q_T is generated by the query rewriting module, it is passed to the query evaluation module. The evaluation and optimization module plays a key role in the security architecture as they are the determinate factors for the efficiency of the framework. Minimally, only the underlying XML document T and the rewritten queries Q_T are needed for evaluating and optimising the queries. However, some more information is essential for efficient evaluation and optimisation. This information could be either at schema level (the XSD D) or at the instance level (e.g., an index).



(a) A visual tool for specifying views

(b) The MFA for Q_0

Figure 7. The user interface of SMOQE

To get the benefit of the instance level optimization, an optional indexer model is needed.

Our system [6] implements a novel algorithm for processing Regular XPath queries represented by MFA. The algorithm, referred to as HyPE (Hybrid Pass Evaluation), takes an MFA as input and evaluates it on an XML tree. A unique feature of HyPE is that it needs a single top-down depth-first traversal of the XML tree, during which HyPE both evaluates predicates of the input query (equivalently, AFA of the MFA) and identifies potential answer nodes (by evaluating the NFA of the MFA). The potential answer nodes are collected and stored in an auxiliary structure, referred to as Cans (candidate answers), which is often much smaller than the XML document tree. After the traversal of the document tree, HyPE only needs a single pass of Cans to select the nodes that are in the answer of the input query. It is important to note that HyPE processes one document tree node at a time, and does not require that the whole document tree is loaded in memory from disk. Therefore, HyPE can be used even in scenarios where the document is not available locally but instead it is streamed over the internet while the query is being evaluated.

To our knowledge, previous systems require to traverse the XML document at least twice to evaluate XPath queries. For example, to evaluate an XPath query q on an XML document T, Arb [12] requires a bottom-up pass of T to evaluate all the predicates of q, followed by a top-down pass to evaluate the selecting path of q. It uses tree automata, which are more complex than MFA and require a pre-processing step (another scan of T) to parse the document and convert it to a special data format (a binary representation of T). In contrast, HyPE is able to evaluate Regular XPath queries, more complex than XPath queries. HyPE evaluator requires neither pre-processing of the data nor the construction of tree automata. It only needs a single pass of the document during which it often prunes a large number of nodes that do not contribute to the answer of the query.

4.4 Result Viewer Module

- **Input:** an XML document T, answer to query Q_T
- **Output:** a text view with XML syntax highlighting, an interactive tree and a schema coloring representing the document T.

The last step of query answering under the security framework is to show the answer of query Q_T (*i.e.*,, the answer of the original query Q_i by user U_i) to the authorized user U_i in the result viewer. The result produced in the query evaluation model is a new XML document. The function of the result viewer model is to present the document in a friendly way to the user. Specifically, the viewer allows three modes: (a) the text mode, which presents the document in a text editor with XML syntax highlighting. (b) the tree mode, which displays the answer of the query as a tree; it provides an interactive interface such that users may click on a node to browse its subtree. (c) the schema mode, which shows the schema graph where the parts of the graph for which the query returned an answer are highlighted.

5 SMOQE: a Reference Implementation

We have developed a reference implementation [7], called SMOQE (Secure MOdular Query Engine), for the security framework we proposed in this paper. It is implemented in Java and supports all the functionality we defined in the framework. It also provides additional functionalities that were not presented previously including:

Manual View Specification. SMOQE does not only support specifying security constraints in the security specification editor and then automatically derive a security view, but also provides the security administrator a more flexible way to enforce security constraints by manually defining a security view. As shown in Figure 7(a), the view specification could be done on a visual view definition tool, which presents administrators with a view schema graph and the query annotation could be filled in by clicking any node in the graph.

More execution modes. SMOQE supports two modes: a DOM mode and a StAX [11] (Streaming API for XML, a new standard API for XML pull parsing, to be included in Java6) mode. In the DOM mode, the whole document tree will be loaded into memory. While in the StAX mode only one sequential scan of the document on disks is needed for the evaluation. The document does not need to be loaded into memory. Comparing to the main-memory XPath engines such as Xalan [19] and Saxon [18], which needs to randomly access the document, the StAX mode enables SMOQE to process larger document efficiently.

Index Management. In SMOQE, an optional index based optimization is supported. In addition, an index management module, as shown in Fig. 8, is implemented, which provides a graphical management interface for maintaining the index.

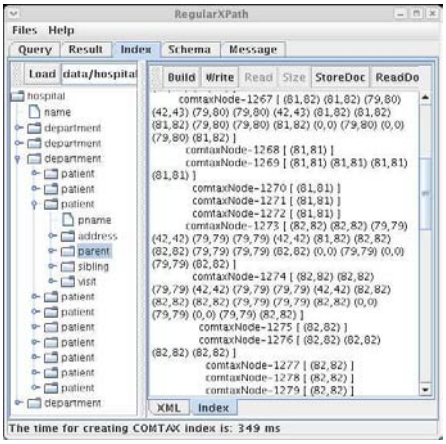


Figure 8. TAX **index**

Monitoring Support. *i*SMOQE is the front-end that not only provides a friendly user interface to SMOQE engine, but also opens a window of the system to let the user monitor the internal processing of the engine. It consists of a graphical querying interface, a semi-automatic view definition tool, and query, automaton, index and result visualization tools.

6 Conclusion

A generic, flexible access control framework for protecting XML data is not only an important research direction, but also a much needed funtionality by numerous XML related applications. In this paper, we have presented a view-based access control framework for XML data and its reference implementation. Taking advantage of a rich security specification language based on XSD and Regular XPath, the framework is able to enforce fine-grained access policies according to the structure and values of the protected XML data. By automatically deriving view specification, the work of security administrators is greatly reduced. By providing proper view schema to different user roles, the protected XML data is no longer a black box, which make it possible to guide users when they pose queries on the system. The automaton based query rewriting, evaluation and optimization modules provide efficient query processing on the derived virtual security views, without any evident degradation in either performance or functionality. Our framework is fully modulized, which leaves space to plug in other view derivation, query rewriting and evaluation techniques.

Several extensions are targeted for future work. First, we are investigating to allow more flexible security specification based on multiple XML data sources as found in data integration. Second, we plan to extend our query rewriting and evaluation modules to handle other XML query languages such as XQuery and XSLT.

References

[1] E. Bertino and E. Ferrari. Secure and selective dissemination of XML documents. *TISSEC*, 5(3):290–331, 2002.
[2] S. Cho, S. Amer-Yahia, L. Lakshmanan, and D. Srivastava. Optimizing the secure evaluation of twig queries. In *VLDB*, 2002.
[3] E. Damiani, S. di Vimercati, S. Paraboschi, and P. Samarati. Securing XML documents. In *EDBT*, 2000.
[4] E. Damiani, S. di Vimercati, S. Paraboschi, and P. Samarati. A fine-grained access control system for XML documents. *TISSEC*, 5(2):169–202, 2002.
[5] W. Fan, C. Y. Chan, and M. Garofalakis. Secure XML querying with security views. In *SIGMOD*, 2004.
[6] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Rewriting regular XPath queries on XML views. *http://www.lfcs.inf.ed.ac.uk/research/database/rewriting.pdf*.
[7] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. SMOQE: A system for providing secure access to xml. In *VLDB*, 2006.
[8] I. Fundulaki and M. Marx. Specifying Access Control Policies for XML Documents with XPath. In *Proc. of the 9th ACM Symposium on Access Control Models and Technologies (SACMAT)*, 2004.
[9] S. Hada and M. Kudo. XML access control language: Provisional authorization for XML documents. `http://www.trl.ibm.com/projects/xml/xacl/xacl-spec.html`.
[10] IBM. DB2 XML Extender. *http://www-3.ibm.com/software/data/db2/extended/xmlext/*.
[11] JSR 173. Streaming API for XML. *http://www.jcp.org/en/jsr/detail?id=173*.
[12] C. Koch. Efficient processing of expressive node-selecting queries on XML data in secondary storage: A tree automata-based approach. In *VLDB*, 2003.
[13] Microsoft. XML support in microsoft SQL server 2005, December 2005. *http://msdn.microsoft.com/library/en-us/dnsql90/html/sql2k5xml.asp/*.
[14] G. Miklau and D. Suciu. Controlling access to published data using cryptography. In *VLDB*, 2003.
[15] M. Murata, A. Tozawa, M. Kudo, and S. Hada. XML access control using static analysis. In *CCS*, 2003.
[16] Oasis. eXtensible Access Control Markup Language (XACML). `http://www.oasis-open.org/committees/xcaml`.
[17] Oracle. Using XML in Oracle internet applications. `http://technet.oracle.com/tech/xml/info/htdocs/otnwp/about_xml.htm`.
[18] SAXON. The XSLT and XQuery processor. *http://saxon.sourceforge.net*.
[19] Xalan. *http://xalan.apache.org*.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Semantic Security in Service Oriented Environments

Mike Surridge, Steve J. Taylor,
E. Rowland Watkins, Thomas Leonard
IT Innovation, Southampton, UK
{ms,sjt,erw,tal}@it-innovation.soton.ac.uk

Terry Payne, Mariusz Jacyno, Ronald Ashri
University of Southampton, UK
{trp,mj04r,ra}@ecs.soton.ac.uk

Abstract

As the technical infrastructure to support Grid environments matures, attention must be focused on integrating such technical infrastructure with technologies to support more dynamic access to services, and ensuring that such access is appropriately monitored and secured. Current approaches for securing organisations through conventional firewalls are insufficient; access is either enabled or disabled for a given port, whereas access to Grid services may be conditional on dynamic factors. This paper reports on the Semantic Firewall (SFW) project, which investigated a policy-based security mechanism responsible for mediating interactions with protected services given a set of dynamic access policies, which define the conditions in which access may be granted to services. The aims of the project are presented, and results and contributions described.

1 Introduction

The Grid Computing paradigm [12] facilitates access to a variety of computing and data resources distributed across geographical and organisational boundaries, thus enabling users to achieve (typically) complex and computationally intensive tasks. In attempting to realise this vision, research and development over recent years has focussed on directing Grid environments towards establishing the fundamentals of the *technical infrastructure* required, as represented by infrastructure development efforts such as the Globus toolkit [13], and standardisation efforts such as OGSA [21] and WS-Resource [9].

However, while such a technical infrastructure is necessary to provide an effective platform to support robust and secure communication, virtualisation, and resource access, other *higher-level* issues need to be addressed before we can achieve the goal of formation and operation of virtual organisations at run-time based on a dynamic selection of services [12]. In particular, whilst low-level security concerns (including encryption, authentication, etc) are addressed, the problems of describing authorised workflows (consisting of the execution of several disparate services) and the policies that are associated with service-access has largely been ignored at this level.

The enforcement of network security policies between different organisations is difficult enough for traditional computing, but becomes even more challenging in the presence of dynamically changing and unpredictable Grid communication needs. Whilst traditional static, network security policies may accommodate the types of traffic required by Grid applications, the same mechanisms can be exploited by crackers for malicious purposes, and thus security policies cannot remain static for long. Firewall policies have long been used as a mechanism to precisely determine what traffic is allowed to flow in and out of an organisational boundary, but require diligent maintenance and monitoring to ensure the integrity of the organisation and avoid breaches of security.

To permit the use of early Grid protocols, firewall policies needed to be relaxed; thus reducing control that system administrators had of their network. In addition, such relaxations could effectively leave a network prone to security breaches, and thus, as security problems arose, ports would subsequently be closed, eliminating the security problems but rendering the corresponding Grid services as useless. Thus, in many cases, some network administrators refuse to open the Grid ports in the first place.

To avoid this problem, the Grid community now tunnels many of its protocols over HTTP(S) using Web Services, exploiting the fact that these transport protocols are more widely used by other communities, and firewall policies are typically more relaxed. Of course, this just moves the target for malicious use, and may leave network administrators with even less control, as now it becomes more difficult to filter Grid traffic without cutting off other services. In ef-

fect, an "arms race" has emerged between Grid developers, malicious users and network administrators, whereby each side has to evolve and change its approach to achieve its desired function. This can result in the need to continuously update Grid applications; such problems are also found in P2P and increasingly also in Web Service applications [19].

In this paper, we present our work on the *Semantic Firewall*, which addresses the conflict between network administrators and Grid (and other Web Service) application developers and users, by:

- analysing the types of exchanges required by typical Grid and Web Service applications[6, 5, 4];
- formulating semantically tractable expressions of the (dynamic) policy requirements for supporting these applications, including access negotiation exchanges where relevant;
- devising message-level authorisation mechanisms that can automatically determine and enforce the applicable (instantaneous) policy for each run-time message exchange;
- implementing a prototype point of control for network administrators, allowing them to manage traffic using high-level dynamic policy rules[15].

This research has been assessed against traditional Grid Application Environments such as GRIA[20] and GEMSS[8], resulting in an understanding of how dynamic security policies could work, how they could be used for security management, and possibly recommendations on the evolution of relevant Semantic Web specifications to include security, as well as prototype software.

This paper summarises the activities and research contributions of the *Semantic Firewall* project, and is structured as follows: Section 2 presents the motivation and high level issues that complicate the task of providing secure access for Grid services, and the architectural components (with justification) of the SFW are presented in Section 3. A description of how the resulting architecture implementation was evaluated is given in Section 4, followed by reflection on the lessons learned in the discussion in Section 5. The paper concludes with Section 6.

2 Motivation

It was clear from the early stages of the project that the *Semantic Firewall (SFW)* is not actually a firewall in the traditional sense (i.e. a perimeter security device), but an active component within an organisation's Grid infrastructure. Thus, the initial investigation focused on mechanisms for supporting dynamic Web Service access control, that

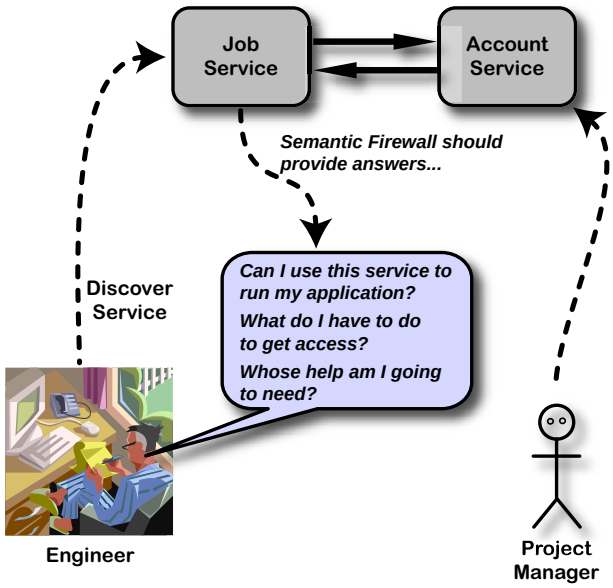


Figure 1. Discovery is just the beginning

could be deployed (and could interact) with the Web Services it was used to protect. Further protection could them be provided by traditional Firewall solutions to secure non-Grid/Web Service traffic; whilst Grid and Web Service traffic would only be routed through the specific Web Service host systems mediated by the deployed Semantic Firewall components. To ensure both pragmatic acceptance and reliability, an essential feature of the SFW architecture is that network administrators have a point of control for Web Service as well as other kinds of traffic.

One of the key issues was the need in Grid applications for users to dynamically discover and exploit services. Whilst this notion works well where the user concerned has permission to use any services they can find, access is often restricted in industrial applications (at least) where their employer has to pay for services. To illustrate this, consider the scenario presented in Figure 1: an industrial Grid application user discovers a job service capable of performing computations needed by the application. However, the job service has a dynamic security policy that prevents access until the user has access to a billing mechanism. In this case, the billing mechanism is provided by an accounting service, which can only be accessed by the user's project manager. The user cannot initially access the job service; however, as the security policy is dynamic, it is possible that they could do so in future, once other actions are taken to negotiate access (i.e. induce a dynamic policy change). These dynamic situations cannot currently be addressed by simply publishing a conventional static policy (e.g. via WS-

Policy[1]).

This scenario highlights many significant issues: the relationship between dynamic security and business (and possibly application) workflows, the fact that a policy may need to express the consequences of interactions between different services, and the need for communication about policy (and its possible future implications) to both users in Figure 1. Some of these issues are similar to those encountered within the Multi-Agent Systems research community[23], and suggest that the SFW should combine Semantic Web, agent and more conventional Web Service security technologies[5].

An emerging, but important issue to consider is that published services are increasingly being developed independently by different providers (for example, different laboratories, organisations, etc), but shared across the public domain, though service description registries, such as UDDI[2]. Whilst the appearance of Grid and web service technologies have facilitated easier access and usage of distributed services for developers, it fails to address many of the knowledge-based problems associated with the diversity of service providers, i.e. interface and data heterogeneity. Unless homogeneity can be assured through a-priori agreed interface and data model specifications, mechanisms to support interoperation between external requests, SFW policies, and the web or Grid service specifications are becoming essential.

The notion of the Semantic Web has been exploited to represent and reason about different services within open, dynamic and evolving environments[7]. The semantic web supports the utilization of many different, distributed ontologies to facilitate reasoning, as well as mappings and articulations that relate correlated concepts within different ontologies. Through the use of reasoning engines, it is possible to infer the relationships between semantically related statements (i.e. statements that are similar in meaning, if not in syntax), and thus bridge the interoperability gap between service representations, queries, and security policies defined by different sources.

Existing work on policies that are based on Semantic Web languages, provide several of the required expressive constructs for defining authorisations and obligations and their delegation. Work such as KAOS[22] takes into account some of the issues relating to conflicting policies between different domains, and provides a centralised means for resolving them. In contrast, Kagal et al. [16] assume a decentralised and adaptive model within REI, whereby the dynamic modification of policies is supported using speech acts and the suggested deployment models for this work examine different scenarios, such as FIPA-compliant agent platforms [3], web pages and web services. However, they do not take into consideration dynamic adaptation of policies within the context of particular interaction scenarios.

An additional challenge is the integration of policy languages with service discovery languages. Denker et. al. have suggested extensions to OWL-S to include policy and security considerations[10], but these typically relate to a single workflow, and do not consider the fact that several actors or services may be required to support dynamic, coordinated access to the type of Grid scenarios presented in Figure 1. Thus, further investigation was necessary to understand how such approaches could be extended to Grid scenarios.

3 Architecture

It is clear from Figure 1 that semantic descriptions of services will be needed, including usage processes with security constraints, so users (or their software agents) can determine whether a service can be used, and if so under what conditions. One solution is simply to represent policy constraints semantically using a policy framework such as KAOS[22] or REI[16], so they can be easily incorporated into a semantic service description[10]. However, this means that semantic inference must be used for run-time policy decisions, which leads to high run-time overheads. A consideration of the policy life-cycle phases reveals three very different requirements in the different phases (Figure 2):

- service descriptions should provide semantically tractable descriptions of behaviour, including any access policy constraints;
- policy enforcement decisions must be made rapidly without any non-trivial semantic inference, to minimise performance overheads which may delay service responses;
- access-policy must be easy to administer, independently of the application services.

3.1 Dynamic process enforcement

Two possible solutions were considered for the dynamic aspects of Semantic Firewall policy determination and enforcement:

- the SFW could monitor messages and use semantic analysis of content to infer whether access should be granted based on the current and previous message; and

[1] http://www-128.ibm.com/developerworks/library/specification/ws-polfram/

[2] http://www.uddi.org/

[3] http://www.fipa.org/

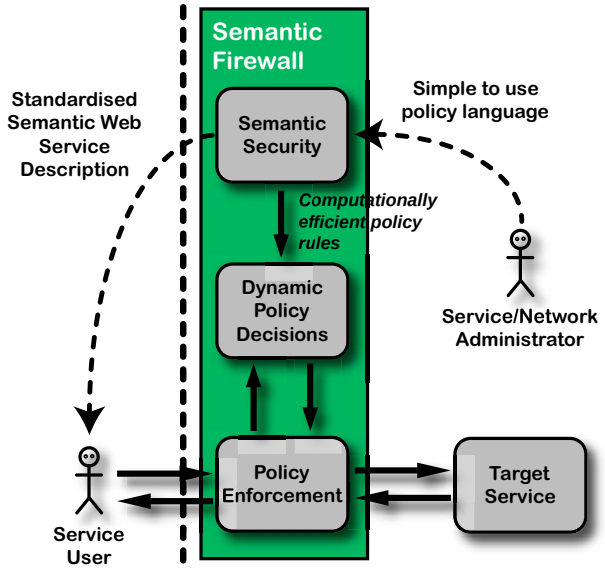


Figure 2. Semantic Firewall Architecture

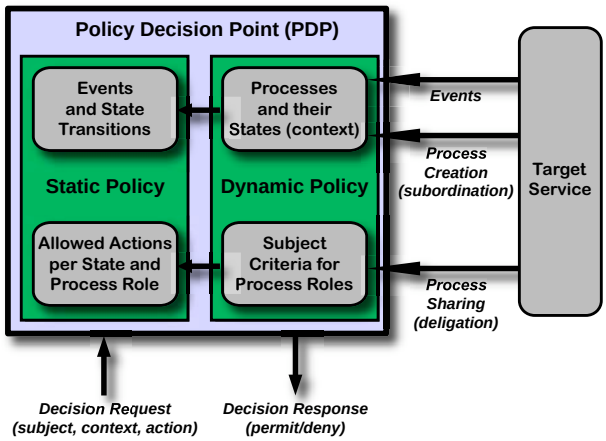


Figure 3. Dynamic Policy Component

- the application services could tell the SFW about state changing events, so the SFW can decide which policies to apply based on the current state.

In practice, message "sniffing" (i.e. monitoring message exchanges between service provider and client) was found to be relatively useless, because in general dynamic policy changes will depend on the logic of the hosted applications, and cannot be inferred purely from the message exchanges unless the whole application logic is reproduced in the Semantic Firewall. Instead, a finite state model definition is necessary that describes the allowed behaviour of an application (one or more related services) in a given interaction between client and service provider (these models are typically used to define the orchestration of a service using such languages as BPEL4WS[4], etc). A particular interaction is identified by a context ID, so-called because it is a reference to the prior context (circumstances occurring previously) of the interaction. Each time a user makes a request, they must quote the context ID, so the service provider knows which interaction the client is talking about. This model includes *rules* about which actions (message exchanges) each type of user can initiate in each state, and which *events* that cause state transitions. Events are messages sent by the target service itself to the SFW, so that the SFW can control which dynamic policy rules are applied to each request, as shown in Figure 3. A transactional model is then needed to specify when policy updates initiated by a service action should take effect. The SFW implements a model in which the consequences of an action on a particular context should be implemented before any other use of that context. Otherwise, race conditions may occur, in that users may be requesting actions based on out of date information.

This approach means the application service developer still encodes the control logic to determine whether a user request was "successful". However, now they must provide a state model describing the expected behaviour of their application (including "unsuccessful" and "erroneous" actions), and generate event messages corresponding to the state model. This state model describes the expected behaviour to the security infrastructure, which can thus enforce it.

To facilitate extensibility, and support organisational policies, the network administrator can also amend the state model, e.g. to insert additional negotiation steps or to reduce the accessible functionality. They can also integrate their own event sources into the original application, e.g. to provide a management function, a negotiation protocol, or a billing service like the account service in Figure 1. Thus, the network administrator can allow applications that require dynamic access rights for remote users, but without losing control over the possible consequences, and without being wholly dependent on the application developer to implement the desired behaviour 100% correctly.

3.2 Process roles and contexts

The permitted actions in each context depend on the state of the application service(s) in that context, and also on the type of the user (as seen by the services) in the corresponding interaction. In the SFW, user types are called *process roles*, since they are defined only with reference to service interactions and need not refer directly to a user's authenticated attributes. When an interaction process starts, a set

[4]http://www-128.ibm.com/developerworks/library/specification/ws-bpel/

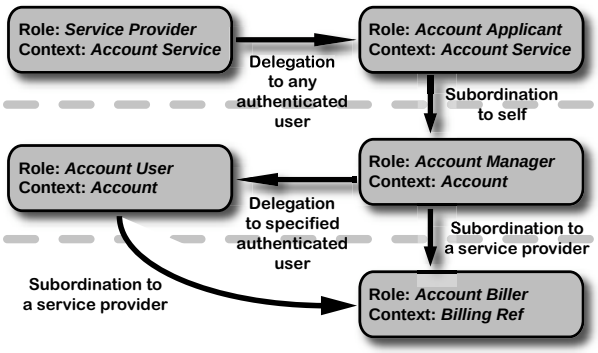


Figure 4. Process roles and contexts: delegation and subordination

of subject criteria are defined for each process role that relates to that interaction normally based on the attributes of the user whose action caused the new process to start. Service actions can then add or remove subject criteria for this or other process roles (a procedure known as *delegation* in the SFW), or initiate new related interactions (a procedure known as *subordination*), as shown in Figure 3.

There are often relationships between processes (contexts) and the user rights associated with them. For example, when the project manager in Figure 1 tries to open an account, they are acting in the role of "account applicant", a right delegated to them by the "service provider" when the service was originally deployed. If the service grants the request, the account becomes a subordinate context in which the successful applicant takes the role of "account manager", and can in turn delegate charging rights (the "account user" process role) to their staff. The full hierarchy is shown in Figure 4.

The notions of process contexts and process roles, and the relationships between them, provides a way to present a high-level interface for generating security policies based on detailed state models of applications. It also provides a way to present a semantically tractable description of the overall policy, using semantic workflow languages such as OWL-S [3] to describe the externally visible processes allowed by the application state models, and RDF or OWL to describe the process roles of interacting actors, and the delegation or subordination relationships between them.

3.3 Security and the Semantic Web

The final step in the Semantic Firewall project is to develop the semantic representations of these dynamic security policies, process roles, etc. It became clear in the final year of the project that the concepts needed arise in both the Semantic Web and the Agent research communities, but they are handled in quite different ways:

Semantic Firewall	Semantic Web	Agents
Application services	WSDL	Scenes & Illocutions
Interaction protocols	OWL-S	Performative Structures
Process roles	—	Allowed Actors
Process contexts	—	—

Table 1. Applicability of Semantic Web and Agents technologies

Initial investigation suggested that Multi-Agent System mechanisms for supporting organisations, such as Electronic Institutions[11] developed at IIIA-CSIC, Spain, can be used to provide the best means to provide tractable descriptions of protected services with interaction policies. The parallels between process roles in the SFW and agent actors has been pursued through a collaboration, resulting in an analysis of the synergies and initial integration between Electronic Institutions and the Semantic Firewall[5].

A deeper analysis showed that, while the parallels are striking, the notion of a process context was far more self-contained in the Electronic Institution interaction models. After some effort to apply these principles to capture process descriptions, it became clear that significant developments would be needed to handle the idea that process contexts can be related. It was also clear that agent models may be difficult to integrate into Grid client applications, which are normally based on workflow models (for example, the Taverna system[18] from myGrid[5]). Given this, the focus moved towards semantic workflow models, and after some investigation of other options such as WSDL-S[6] and WSMO[7], the simpler but more mature OWL-S specification[3] was selected.

The main challenge was to introduce the process contexts and roles into OWL-S, and to semantically encode the permissions associated with the process roles in a given process context. It was immediately clear that OWL-S can describe a generic "process", including the notion of a "process state" that affects which actions can be taken. However, the process model as defined in the W3C submission[8] fails to support the notion that processes can be attributed to specific contexts that can be dynamically created, evolve, and be destroyed, and about which queries can be posed that should be resolvable using the generic description. As the focus for the Semantic Firewall project is to define and support dynamic security, the more general problem of such dynamism (including service discovery, provisioning, com-

[5] http://www.mygrid.org.uk/
[6] http://www.w3.org/Submission/WSDL-S/
[7] http://www.w3.org/Submission/WSMO/
[8] http://www.w3.org/Submission/2004/07/

position, etc.) were not addressed in this project. Rather, each process context was described as a distinct and separate service: for example, the transfer of data, managing off-site storage, and handling different billing accounts. Thus, in order to utilise a particular context, the service client simply uses the service description defined for a particular context.

This left only the process roles and their access rights to be added to OWL-S. This was done by defining a set of pre- and post-conditions, as follows:

1. the process state(s) from which a supported workflow could be started were encoded as pre-conditions to the corresponding OWL-S process description;
2. the process role(s) required to enact each workflow were also included as OWL-S pre-conditions;
3. the outcome (goal achieved) by each workflow was included as an OWL-S process post-condition;

Where appropriate, the process-role pre-conditions were accompanied by a "qualification" action, through which the actor could be granted that process role. This was encoded as a goal that would have to be achieved by finding and executing another workflow in the same context, whose outcome would be to grant the required role. Note that this second workflow also had process role pre-conditions, and typically could not be executed by the original actor. However, the pre-conditions could then be used to look for another actor (e.g. a supervisor) who could execute the required workflow.

The end result was an enrichment of the basic OWL-S process descriptions, in which the security requirements to execute each process are described, along with mechanisms that could be used (if authorised) to acquire the necessary access rights. These mechanisms typically point to other users who have control of parent contexts (for example, a prospective service user needs to acquire access permission from the manager of the account to which the service bills).

4 Validation

To validate the Semantic Firewall approach, we needed a Grid (or Web Services) application in which policies can change dynamically, e.g. as a result of a negotiation procedure. As suggested by Figure 1, such applications are typically found in inter-enterprise business Grids, such as those originally created by the GRIA[9][20] and GEMSS[10][8] projects. These Grids are forced to use at least some dynamic policy elements to handle changing business relationships, and in some cases legal constraints over the movement of data [14].

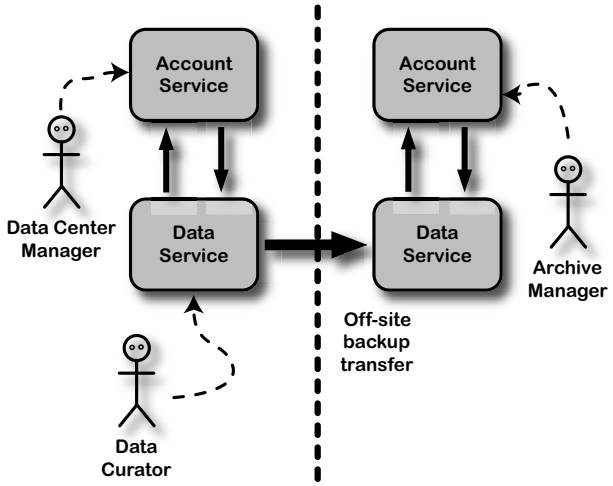


Figure 5. A simple test scenario

The validation study was conducted, using an alpha release of the forthcoming GRIA 5 release. This uses dynamic token services developed in the NextGRID project [1, 17] and exhibits a wider range of dynamic behaviour than the original GEMSS or GRIA systems. GRIA 5 will also use the dynamic policy decision component from the Semantic Firewall itself, which meant it was easy to integrate with a Semantic Firewall deployment for evaluation tests.

Because the GRIA 5 system is so dynamic, it was possible to test the Semantic Firewall principles even with a very simple application. The scenario involves a data centre curator transferring a large data file between two sites, e.g. as part of a resilient back up strategy. The transfer is conducted using a GRIA 5 data service at each site, but these services will also bill the user for their actions, using account services located at each site. The relevant accounts are managed by other actors, e.g. the manager of the data centre where the curator works, and the manager of the network storage archive service providing the off-site backup facility. The services and actors are shown in Figure 5.

The focus of the validation exercise was on whether the SFW approach could accommodate semantically tractable service descriptions, and whether these could be used to determine if and how the curator could achieve their data transfer goal. Implementation of the application scenario was trivial using the GRIA 5 system, and registry services were added to store information about the available services (including their semantic descriptions), and information about users and their roles. Note that the user information available to each actor was incomplete the registry

[9] GRIA is now an open source Grid middle-ware, currently at v4.3, obtainable via http://www.gria.org

[10] http://www.gemss.de

revealed user roles only to those actors for whom the user was willing to exercise those roles, i.e. the registry filtered according to the trust relationships between users.

To validate the use of semantic descriptions, the MindSwap OWL-S API and execution engine[11] were used to execute the required task, based on OWL-S workflows discovered from the service and workflow registry. Reasoning over the pre-conditions allowed classification of discovered services and workflows into three classes: those that could be executed by the user, those that could become executable by the user with help from other actors, and those that could never be executed. In the second case, the reasoning engine was able to work out how to negotiate the required access rights with the SFW infrastructure, with help from other actors (e.g. a supervisor).

In the final experiments, software agents (services) were created to represent each actor, which would take in a workflow (or goal) from other users, and (where trustworthy) enact the workflow for that user. With these services in place, it was possible to automate the negotiation process, thus exploiting the parallels and synergies between Semantic Web, agents, and dynamic security.

5 Discussion

The investigation of the Semantic Firewall to date has demonstrated that security policies can be defined for semantically described process models, with little additional run-time (semantic reasoning) overhead[15]. The research has highlighted a number of challenges for supporting dynamic, context-specific service access. The use of policies introduces a degree of flexibility for both service providers and for the network administrators. By representing these policies as state models (provided by the service providers themselves), network administrators can provide their own extension or modifications as necessary. To ensure that conflicts or violations do not occur between service and system policies, and to provide pragmatic tools to manage policy definitions, further investigation is still required.

An important consideration is that of managing policy violations, and mediating the type of response that can assist clients who are attempting to make legitimate service requests, whilst avoiding the exposure of policy details that would enable malicious users to bypass the security mechanisms. By currently exposing the defined policies, clients can reason about candidate workflows to determine whether they will fail, and thus avoid committing to workflows that would subsequently need revising. In addition, the current policy definitions allow the inference of other, necessary stages (and consequently actors) that would assist in establishing the appropriate access rights.

[11] http://www.mindswap.org/2004/owl-s/api/

The decision to use a semantic representation for the SFW presented a challenge given that to associate the policies defined by the Semantic Firewall with workflows, the service description language should be extended. The OWL-S ontologies provided the ideal platform for such extensions, and in [15] we present the set of OWL-S extensions proposed. In addition, the execution environment had to be extended to support the notion of process abstractions (through OWL-S *simple processes*) that could subsequently be provisioned at run time by consulting local service and user registries (typically within the same Virtual Organisation as the service client). For example, a client might realise that they need to acquire credentials from their local account manager before accessing those services with which such prior agreements had been made.

The research is ongoing, and future work will address extending the OWL-S extensions further, to support other notions of service. Akkermans et al. [2] introduced such notions as service bundling and the sharability and consumability of resources within the OBELIX project. Current semantic web service descriptions fail to make the distinction, for example, between resources that can be copied and shared by many users, and that which is indivisible (e.g. a security credential that can only be used by one user at the time). Likewise, there is no support for establishing relationships between service elements that support each other, but are not necessarily part of a service workflow (such as representing a billing service that supports another, primary service). Future investigation will consider how such factors augment the definition of Grid services and further support policy definitions.

6 Conclusions

In this paper, we have summarised research conducted as part of the EPSRC eScience funded *Semantic Firewall* project. The problem of providing secure access to services was analysed within a Grid scenario, which identified several challenges not normally apparent when considering simple workflows. A semantically-annotated policy model has been developed, that supports the definition of permissible actions/workflows for different actors assuming different roles for given processes. An initial prototype implementation has been developed, which has been used to validate the model on real-world, Grid case studies, and several areas for future work have been identified.

7 Acknowledgment

This research is funded by the Engineering and Physical Sciences Research Council (EPSRC) Semantic Firewall project (ref. GR/S45744/01). We would like to acknowledge contributions made by Grit Denker (SRI) for valuable

insights into the requirements for semantically annotated policies and interaction workflows. Thanks are also due to Jeff Bradshaw and his research group for valuable discussions on the use and possible applicability of the KaOS framework to Grid Services, and to Carles Sierra and his group on understanding how notions of Electronic Institutions could be exploited in defining and deploying a Semantic Firewall framework.

References

[1] M. Ahsant, M. Surridge, T. Leonard, A. Krishna, and O. Mulmo. Dynamic Trust Federation in Grids. In *the 4th International Conference on Trust Management (iTrust 2006)*, Pisa, Italy, 2006.

[2] H. Akkermans, Z. Baida, J. Gordijn, N. Pena, A. Altuna, and I. Laresgoiti. Value Webs: Using Ontologies to Bundle Real-World Services. *IEEE Intelligent Systems*, 19(4):57–66, 2004.

[3] A. Ankolekar, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, and K. Sycara. DAML-S: Web Service Description for the Semantic Web. In *First International Semantic Web Conference (ISWC) Proceedings*, pages 348–363, 2002.

[4] R. Ashri, G. Denker, D. Marvin, M. Surrdige, and T. R. Payne. Semantic Web Service Interaction Protocols: An Ontological Approach. In S. A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *Int. Semantic Web Conference*, volume 3298 of *LNCS*, pages 304–319. Springer, 2004.

[5] R. Ashri, T. Payne, M. Luck, M. Surridge, C. Sierra, J. A. R. Aguilar, and P. Noriega. Using Electronic Institutions to secure Grid environments. In *10th International Workshop on Cooperative Information Agents (Submitted)*, 2006.

[6] R. Ashri, T. Payne, D. Marvin, M. Surridge, and S. Taylor. Towards a Semantic Web Security Infrastructure. In *Proceedings of Semantic Web Services 2004 Spring Symposium Series*, 2004.

[7] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):35–43, 2001.
Essay about the possibilities of the semantic web.

[8] G. Berti, S. Benkner, J. W. Fenner, J. Fingberg, G. Lonsdale, S. E. Middleton, and M. Surridge. Medical simulation services via the grid. In S. Nørager, J.-C. Healy, and Y. Paindaveine, editors, *Proceedings of 1st European HealthGRID Conference*, pages 248–259, Lyon, France, Jan. 16–17 2003. EU DG Information Society.

[9] K. Czajkowski, D. F. Ferguson, F. I, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, and W. Vambenepe. The WS-Resource Framework. Technical report, The Globus Alliance, 2004.

[10] G. Denker, L. Kagal, T. Finin, M. Paolucci, and K. Sycara. Security for DAML Services: Annotation and Matchmaking. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *Proceedings of the 2nd International Semantic Web Conference*, volume 2870 of *LNCS*, pages 335–350. Springer, 2003.

[11] M. Estena. *Electronic Institutions: from specification to development*. PhD thesis, Technical University of Catalonia, 2003.

[12] I. Foster and C. Kesselman. *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2003.

[13] I. Foster, C. Kesselman, J. M. Nick, and S. Tuecke. Grid Services for Distributed System Integration. *IEEE Computer*, 35(6):37–46, June 2002.

[14] J. Herveg, F. Crazzolara, S. Middleton, D. Marvin, and Y. Poullet. GEMSS: Privacy and security for a Medical Grid. In *Proceedings of HealthGRID 2004*, Clermont-Ferrand, France, 2004.

[15] M. Jacyno, E.R.Watkins, S. Taylor, T. Payne, and M. Surridge. Mediating Semantic Web Service Acces using the Semantic Firewall. In *The 5th International Semantic Web Conference (Submitted)*, 2006.

[16] L. Kagal, T. Finin, and A. Joshi. A Policy Based Approach to Security for the Semantic Web. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *2nd Int. Semantic Web Conference*, volume 2870 of *LNCS*, pages 402–418. Springer, 2003.

[17] T. Leonard, M. McArdle, M. Surridge, and M. Ahsant. Design and implementation of dynamic security components. NextGRID Project Output P5.4.5, 2006.

[18] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. C. adn K. Glover, M. Pocock, A. Wipat, and P. Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.

[19] M. Surridge and C. Upstill. Lessons for Peer-to-Peer Systems. In *Proceedings of the 3rd IEEE Conference on P2P Computing*, 2003.

[20] S. Taylor, M. Surridge, and D. Marvin. Grid Resources for Industrial Applications. In *2004 IEEE Int. Conf. on Web Services (ICWS'2004)*, 2004.

[21] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, D. Snelling, and P. Vanderbilt. Open grid services infrastructure. Technical report, Global Grid Forum, 2003.

[22] A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. J. Hayes, M. R. Breedy, L. Bunch, M. Johnson, S. Kulkarni, and J. Lott. KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement. In *4th IEEE Int. Workshop on Policies for Distributed Systems and Networks*, pages 93–98. IEEE Computer Society, 2003.

[23] M. J. Wooldridge. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Formal Analysis of Access Control Policies

Jeremy W. Bryans
School of Computing Science
Newcastle University
UK

Abstract

We present a formal (model-based) approach to describing and analysing access control policies. This approach allows us to evaluate access requests against policies, compare versions of policies with each other and check policies for internal consistency. Access control policies are described using VDM, a state-based formal modelling language. Policy descriptions are concise and may be easily manipulated. The structure of the VDM description is derived from the OASIS standard access control policy language XACML. It is therefore straightforward to translate between XACML policies and their corresponding models.

1 Introduction

In many market sectors, business affiliations are increasingly volatile. Use of technologies such as web services can allow companies to rapidly join forces in *virtual organisations* (VOs), tailored to meet specific market opportunities. The structure of these VOs may be constantly changing, as companies join to provide necessary new skills, or members complete their part of a task and withdraw from the VO.

In order for the VO to operate efficiently, each partner must expose relevant business information to the other members. This may be done using a centralised database, to which all partners contribute, or it may be done by each partner retaining their own business information and allowing other partners access as necessary.

Parts of the overall access control policy may be developed and maintained by each partner, and it may be that no one person is responsible for the final resultant policy. Either way, the relevant access control policies need to be updated carefully, to ensure that legitimate accesses are not blocked (which would slow down the operation of the VO) and that illegitimate accesses are not permitted. In this environment there is a clear need to understand the behaviour of the policy and in particular to understand how changes to small parts of the policy will affect the behaviour of the overall policy.

The GOLD project [14] has been researching into enabling technology to support the formation, operation and termination of VOs. In this paper we present a formal way of analysing access control policies which have been written in XACML. We translate policies into the formal modelling language VDM. Simple operations within this formal context permit us to test the behaviour of these policies, compare policies with each other and check the internal consistency of policies. The structure of the XACML is preserving and therefore a faulty policy can be fixed within the VDM framework and translated back to XACML.

VDM is a model oriented formal method with good tool support: VDMTools [3].

In Section 2 we give an overview of XACML, the OASIS access control language. Section 3 gives a VDM description of the data types and algorithms common to all access control policies. Section 4 shows how these data types may be populated to describe a specific policy. In Section 5 we show how the VDMTools framework may be used to test policies, and

compare them with each other.

2 XACML and access control

XACML [13] is the OASIS standard for access control policies. It provides a language for describing access control policies, and a language for interrogating these policies, to ask the policy if a given action should be allowed.

A simplified description of the behaviour of an XACML policy is as follows. An XACML policy has an associated *Policy Decision Point* (PDP) and a *Policy Enforcement Point* (PEP) (See Figure 1.) Any access requests by a user are intercepted by the PEP. Requests are formed in the XACML request language by the PEP and sent to the PDP. The PDP evaluates the request with respect to the access control policy. This response is then enforced by the PEP.

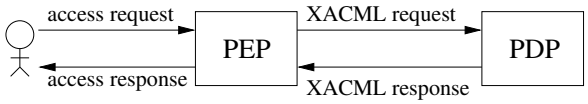


Figure 1: XACML overview.

When a request is made, the PDP will return exactly one of:

- PERMIT: if the subject is permitted to perform the action on the resource,
- DENY: if the subject is not permitted to perform the action on the resource, or
- NOTAPPLICABLE: if the request cannot be answered by the service.

The full language also contains the response INDETERMINATE. This is triggered by an error in evaluating the conditional part of the rule. Since we assume rules to be environment independent, evaluating the rules will not require evaluation of any conditional statements. We therefore do not use INDETERMINATE.

A full XACML request includes a set of *subjects* (e.g. a user, the machine the user is on, and the applet the user is running could all be subjects with varying access rights) to perform a set of *actions* (e.g. read, write, copy) on a set of *resources* (e.g. a file or a disk) within an environment (e.g. during work hours or from a secure machine). It may also contain a *condition* on the environment, to be evaluated when the request is made. We will assume that requests (and later, rules) are environment independent, and omit the condition and environment components.

We will assume that a PDP contains a set of *Policies*, each of which contain a set of *Rules*[1]. Rules in XACML contain a *target* (which further contains sets of resources, subjects and actions) and an *effect* (permit or deny). If the target of a rule matches a request, the rule will return its *effect*. If the target does not match the rule NOTAPPLICABLE is returned.

As well as a set of rules, a policy contains a *rule combining algorithm*. Like rules, they also contain a target. All rules within a policy are evaluated by the PDP. The results are then combined using the rule combining algorithm, and a single effect is returned.

The PDP evaluates each policy, and combines the results using a *policy combining algorithm*. The final effect is then returned to the PEP to be enforced. If PERMIT is returned then the PEP will permit access, any other response will cause access to be denied.

3 VDM

VDM is a model oriented formal method incorporating a modelling or specification language (VDM-SL) with formal semantics [1], a proof theory [2] and refinement rules [11].

A VDM-SL model is based on a set of data type definitions, which may include invariants (arbitrary predicates characterising properties shared by all members of the type). Functionality is described in terms of functions over the types, or operations which may have side effects on distinguished state variables. Functions and operations may be restricted by preconditions, and may be defined in an explicit algorithmic style or implicitly in terms of postcondi-

[1] Strictly, it contains a set of policy sets, each of which contain a set of policies. Policies then contain a set of rules. Extending our model to include this additional complexity would be straightforward.

tions. The models presented in this paper use only explicitly-defined functions. We remain within a fully executable subset of the modelling language, allowing our models of XACML policies to be analysed using an interpreter.

VDM has strong tool support. The CSK VDM-Tools [3] include syntax and type checking, an interpreter for executable models, test scripting and coverage analysis facilities, program code generation and pretty-printing. These have the potential to form a platform for tools specifically tailored to the analysis of access control policies in an XACML framework.

An access control policy is essentially a complex data type, and the XACML standard is a description of the evaluation functions with respect to these data types. Thus VDM, with its focus on datatypes and functionality, is a suitable language to describe access control policies.

This section describes in detail the data types and functionality of the Policy Decision Point. The description is presented in VDM. We impose the simplifications mentioned in the previous section. In particular, we limit targets to sets of subjects, actions and resources, and exclude consideration of the environment. This means we only consider the effects PERMIT, DENY and NOTAPPLICABLE.

Elements of the type *PDP* are pairs. The first field has label *policies* and contains a set of the elements of the type *Policy*. The second field contains one value of the type *CombAlg*, defined immediately below.

$$\begin{array}{rrcl} PDP :: & policies & : & Policy\text{-}\mathbf{set} \\ & policyCombAlg & : & CombAlg \end{array}$$

$$CombAlg = \textsc{DenyOverrides} \mid \textsc{PermitOverrides}$$

DENYOVERRIDES and PERMITOVERRIDES are enumerated values. They will act as pointers to the appropriate algorithms, which are defined later. Other possible combining algorithms are given in [13] but for simplicity we will model only these two here.

A policy contains a *target* (the sets of *subjects*, *resources* and *actions* to which it applies), a set of *rules*, and a *rule combining algorithm*.

$$\begin{array}{rrcl} Policy :: & target & : & Target \\ & rules & : & Rule\text{-}\mathbf{set} \\ & ruleCombAlg & : & CombAlg \end{array}$$

$$\begin{array}{rrcl} Target :: & subjects & : & Subject\text{-}\mathbf{set} \\ & resources & : & Resource\text{-}\mathbf{set} \\ & actions & : & Action\text{-}\mathbf{set} \end{array}$$

Each rule has a target and an *effect*. If a request corresponds to the rule target then the rule effect is returned. The brackets [..] denote that the target component may be empty. In this case, the default value is the target of the enclosing policy.

$$\begin{array}{rrcl} Rule :: & target & : & [Target] \\ & effect & : & Effect \end{array}$$

The effect of the rule can be PERMIT, DENY, or NOTAPPLICABLE. These are modelled as enumerated values.

$$Effect = \textsc{Permit} \mid \textsc{Deny} \mid \textsc{NotApplicable}$$

Requests are simply *target*s:

$$Request :: target : Target$$

The functionality of a PDP is captured in the following set of functions. We begin with the function *targetmatch*, which takes two targets and returns true if the targets have some subject, resource and action in common.

$$targetmatch : Target \times Target \to Bool$$

$$\begin{array}{l} targetmatch(target1, target2) \triangleq \\ \quad (target1.subjects \cap target2.subjects) \neq \{\} \wedge \\ \quad (target1.resources \cap target2.resources) \neq \{\} \wedge \\ \quad (target1.actions \cap target2.actions) \neq \{\} \end{array}$$

A request is evaluated against a rule using *evaluateRule*. If the rule target is NULL the targets are assumed to match, since the parent policy target must match. If the targets match the effect of the rule is returned, otherwise NOTAPPLICABLE is returned.

$$evaluateRule : Request \times Rule \to Effect$$

$$\begin{array}{l} evaluateRule(req, rule) \triangleq \\ \quad \textbf{if } rule.target = \textsc{Null} \\ \quad \textbf{then } rule.effect \\ \quad \textbf{else if } targetmatch(req.target, rule.target) \\ \qquad \textbf{then } rule.effect \\ \qquad \textbf{else } \textsc{NotApplicable} \end{array}$$

A policy is invoked if its target matches the request. It then evaluates all its rules with respect to a request, and combines the returned effects using its *rule combining algorithm*. In the following DENYOVERRIDES is abbreviated to D-OVER, and PERMITOVERRIDES to P-OVER.

evalPol : *Request* × *Policy* → *Effect*

evalPol(*req*, *pol*) $\triangleq$
  if *targetmatch*(*req.target*, *pol.target*)
  then if (*pol.ruleCombAlg* = D-OVER)
    then *evalRules-DO*(*req*, *pol.rules*)
    else if (*pol.ruleCombAlg* = P-OVER)
      then *evalRules-PO*(*req*, *pol.rules*)
      else NOTAPPLICABLE
  else NOTAPPLICABLE

The deny overrides algorithm is implemented as

evalRules-DO : *Request* × *Rule*-**set** → *Effect*

evalRules-DO(*req*, *rs*) $\triangleq$
  if $\exists r \in rs \cdot evaluateRule(req, r) = \text{DENY}$
  then DENY
  else if $\exists r \in rs \cdot evaluateRule(req, r) = \text{PERMIT}$
    then PERMIT
    else NOTAPPLICABLE

If any rule in the policy evaluates to DENY, the policy will return DENY. Otherwise, if any rule in the policy evaluates to PERMIT, the policy will return PERMIT. If no rules evaluate to either PERMIT or DENY, the policy will return NOTAPPLICABLE.

The permit overrides rule combining algorithm (omitted) is identical in structure, but a single PERMIT overrides any number of DENYS.

The evaluation of the PDP and its rule combining algorithms has an equivalent structure to the policy evaluation functions already presented.

evaluatePDP : *Request* × *PDP* → *Effect*

evaluatePDP(*req*, *pdp*) $\triangleq$
  if (*pdp.policyCombAlg* = D-OVER)
  then *evalPDP-DO*(*req*, *pdp*)
  else if (*pdp.policyCombAlg* = P-OVER)
    then *evaluatePDP-PO*(*req*, *pdp*)
    else NOTAPPLICABLE

evaluatePDP-DO : *Request* × *PDP* → *Effect*

evaluatePDP-DO(*req*, *pdp*) $\triangleq$
  if $\exists p \in pdp.policies \cdot evalPol(req, p) = \text{DENY}$
  then DENY
  else if $\exists p \in pdp.policies \cdot evalPol(req, p) = \text{PERMIT}$
    then PERMIT
    else NOTAPPLICABLE

The above functions and data types are generic. Any XACML policy in VDM will use these functions. In the next section we show how to instantiate this generic framework with a particular policy.

4 An example policy

In this section we present the initial requirements on an example policy and instantiate our abstract framework with a policy aimed at implementing these requirements. In Section 5 we show how we can use the testing capabilities of VDMTools [3] to find errors in these policies.

This example is taken from [5], and describes the access control requirements of a university database which contains student grades. There are two types of resources (*internal* and *external* grades), three types of *actions* (*assign*, *view* and *receive*), and a number of subjects, who may hold the roles *Faculty* or *Student*. We therefore define

Action = ASSIGN | VIEW | RECEIVE

Resource = INT | EXT

Subjects are enumerated values,

Subject = ANNE | BOB | CHARLIE | DAVE

and we populate the Student and Faculty sets as

Student : *Subject*-**set** = {ANNE, BOB}
Faculty : *Subject*-**set** = {BOB, CHARLIE}

so Bob is a member of both sets. In practice, an access control request is evaluated on the basis of certain attributes of the subject. What we are therefore saying here is that the system, if asked, can produce evidence of Anne and Bob being students, and of Bob and Charlie being faculty members.

Informally, we can argue that populating the student and faculty sets so sparsely is adequate for testing purposes. All rules we go on to define apply to roles, rather than to individuals, so we only need one representative subject holding each possible role combination.

The properties to be upheld by the policy are

1. No students can assign external grades,
2. All faculty members can assign both internal and external grades, and
3. No combinations of roles exist such that a user with those roles can both receive and assign external grades.

Our initial policy (following the example in [5]) is

> *Requests for students to receive external grades, and for faculty to assign and view internal and external grades, will succeed.*

Implementing this policy naïvely leads to the following two rules, which together will form our initial (flawed) policy. Students may receive external grades,[2]

$$StudentRule : Rule = ((Student, \text{EXT}, \text{RECEIVE}), \text{PERMIT})$$

and faculty members may assign and view both internal and external grades.

$$FacultyRule : Rule = ((Faculty, \{\text{INT}, \text{EXT}\}, \{\text{ASSIGN}, \text{VIEW}\}), \text{PERMIT})$$

The policy combines these two rules using the PERMITOVERRIDES algorithm. The target of the policy is all requests from students and faculty members.

$$PolicyStuFac : Policy = ((Student \cup Faculty, \{\text{INT}, \text{EXT}\}, \{\text{ASSIGN}, \text{VIEW}, \text{RECEIVE}\}), \{StudentRule, FacultyRule\}, \text{P-OVER})$$

[2]The correct VDM description is

$$StudentRule : Rule = mk\_Rule(mk\_Target(Student, \{\text{EXT}\}, \{\text{RECEIVE}\}), \text{PERMIT})$$

For ease of reading, we omit the $mk\_$ constructs and brackets around singleton sets.

In fact, this policy would have the same behaviour if the two rules were combined using the DENYOVERRIDES algorithm. This is because both rules have the effect PERMIT.

The PDP is a collection of policies; in this case only one.

$$PDPone : PDP = (PolicyStuFac, \text{D-OVER})$$

We use the deny overrides algorithm here, but in this case it has the same behaviour as permit overrides, because there is only one policy. In the next section we examine this and other specifications using VDM-Tools.

5 Testing the specification

The VDM toolset VDMTools described in [7] provides considerable support for testing VDM specifications, including syntax and type checking, testing and debugging. Individual tests can be run at the command line in the interpreter. The test arguments can be also read from pre prepared files, and scripts are available to allow large batches of tests to be performed.

A systematic approach to testing requires that we have some form of oracle against which to judge the (in)correctness of test outcomes. This may be a manually-generated list of outcomes (where we wish to assess correctness against expectations) or an executable specification (if we wish to assess correctness against the specification). In this section we use a list of expected results as our oracle. Section 5.1 uses one version of a VDM-SL specification as an oracle against another version.

Tests on *PDPone* are made by forming requests and evaluating the PDP with respect to these requests, using the function *evaluatePDP* from Section 3.

Below we show four example requests and the results from *PDPone*.

Request	Result from *PDPone*
(ANNE, EXT, ASSIGN)	NOTAPPLICABLE
(BOB, EXT, ASSIGN)	PERMIT
(CHARLIE, EXT, ASSIGN)	PERMIT
(DAVE, EXT, ASSIGN)	NOTAPPLICABLE

In the first test, *PDPone* returns NotApplicable when user Anne (a student) asks to assign an external grade. This is because there is no rule which specifically covers this situation. The PEP denies any request which is not explicitly permitted, and so access is denied.

The second test points out an error, because Bob (who is both a faculty member and a student) is allowed to assign external grades, in violation of rule one, which states that no student may assign external grades. This policy has been written with an implicit assumption that the sets student and faculty are disjoint. Constraining these sets to be disjoint when we populate them allows us to reflect this assumption. In practice this constraint would have to be enforced at the point where roles are assigned to individuals rather than within the PDP.

The third test is permitted, as expected, since Charlie is a member of faculty, and the fourth test returns NotApplicable, because Dave is not a student or a faculty member.

Multiple requests

The policy as defined can be broken if multiple access control requests are combined into one XACML request. For example the request below (identified in [5])

(Anne, {Ext}, {Assign, Receive})

is permitted. As pointed out in [5], this breaks the first property, because Anne (a student) is piggybacking an illegal request (assigning an external grade) on a legal one (receiving an external grade). In future, therefore, we make the assumption that the PEP only submits requests that contain singleton sets. Given this assumption, we can limit the test cases we need to consider to those containing only single subjects, actions and resources.

5.1 Comparing specifications of PDPs

We now suppose (following [5]) that teaching assistants (TAs) are to be employed to help with the internal marking. They are not, however, allowed to help with external marking. A careless implementation, that merely included the names of the TAs as faculty members, would overlook the fact that students are often employed as TAs.

A more robust implementation, that makes TAs a separate role and develops rules specific for them, is given below. Note that in *TArule*2, TAs are explicitly forbidden to assign or view external grades; their role is restricted to dealing with the internal grades.

$$TArule1 : Rule = ((TA, \{\text{Int}\}, \{\text{Assign}, \text{View}\}), \text{Permit})$$

$$TArule2 : Rule = ((TA, \{\text{Ext}\}, \{\text{Assign}, \text{View}\}), \text{Deny})$$

The rules are combined into a (TA-specific) policy

$$PolicyTA: Policy = ((TA, \{\text{Int}, \text{Ext}\}, \{\text{Assign}, \text{View}, \text{Receive}\}), \{TArule1, TArule2\}, \text{PermitOverrides})$$

which is combined with *PolicyStuFac* from Section 4 to give a new Policy Decision Point:

$$PDPtwo : PDP = (\{PolicyTA, PolicyStuFac\}, \text{D-Over})$$

This new PDP can of course be tested independently, but it can also be compared with the previous one. We do this with respect to a test suite. Because the policies are small, this test suite can be comprehensive. We now populate the roles as

$$\begin{array}{lcl} Student : Subject\text{-}\mathbf{set} & = & \{\text{Anne}, \text{Bob}\} \\ Faculty : Subject\text{-}\mathbf{set} & = & \{\text{Charlie}\} \\ TA : Subject\text{-}\mathbf{set} & = & \{\text{Bob}, \text{Dave}\} \end{array}$$

taking care that there is a person holding each possible combination of roles that we allow. Every request that each person can make is considered against each PDP. This is easily automated using a simple shell script. The observed changes are summarised below.

Request	*PDPone*	*PDPtwo*
(Bob, Int, Assign)	NotApp	Permit
(Bob, Int, View)	NotApp	Permit
(Bob, Ext, Assign)	NotApp	Deny
(Bob, Ext, View)	NotApp	Deny
(Dave, Int, Assign)	NotApp	Permit
(Dave, Int, View)	NotApp	Permit
(Dave, Ext, Assign)	NotApp	Deny
(Dave, Ext, View)	NotApp	Deny

As a *TA*, Bob's privileges now include assigning and viewing internal grades, as well as all the privileges he has as a student. Everything else is now explicitly denied.

All requests from Dave, who is a now TA but not a student, are judged NotApplicable (and consequently denied) by the first policy, but the second policy allows him to view and assign internal grades. It explicitly forbids him to assign or view external grades.

Internal consistency of a PDP

We consider a set of rules to be consistent if there is no request permitted by one of the rules which is denied by another in the set. A set of policies is consistent if there is no request permitted by one of the policies which is denied by another in the set.

Rule consistency within a policy and policy consistency within a PDP can each be checked using the method outlined above, using the functions *evaluateRule* and *evaluatePol* from Section 3.

6 Related Work

An important related piece of work (and indeed a major source of inspiration for this work) is [5]. Here the authors present Margrave, a tool for analysing policies written in XACML. Our intentions are almost identical but there are some important differences. Margrave transforms XACML policies into Multi-Terminal Binary Decision Diagrams (MTBDDs) and users can query these representations and compare versions of policies. Our work allows the possibility of the user manipulating the VDM representation of a policy then translating the changed version back into XACML. We test policies against requests where Margrave uses verification.

In [8, 15] the access control language RW is presented. It is based on propositional logic, and tools exist to translate RW programs to XACML, and to verify RW programs.

Alloy [10] has been used [9] to verify access control policies. XACML policies are translated into the Alloy language and partial ordering between these polices can be checked.

7 Conclusions and Further Work

We have presented a formal approach to modelling and analysing access control policies. We have used VDM, a well-established formal method, as our modelling notation. This has allowed us to use VDMTools to analyse the resultant formal models. We have shown that rigorous testing of these policies is possible within VDMTools, and further that policies may be checked for internal consistency.

Ongoing work seeks to represent rules that are dependent on context. This will require extending the VDM model with environmental variables, and allowing rules to query these variables. This will allow us to model a much broader range of policies, including *context-based authorisation* [4] and delegation.

With larger policies, testing all possible requests may become time consuming. Further work will look at techniques and tools for developing economical test suites for access control policies.

Using attributes of subjects instead of subject identities would allow a greater range of policies to be implemented, and this is something we are currently working on.

A full implementation of the translation from XACML to VDM and vice versa is under development.

Delegation could then also be modelled. The delegator could alter a flag in the environment (perhaps by invoking a certain rule in the PDP) and the delegate is then only allowed access if the flag is set.

Following the RBAC profile of XACML [12], the

rules we have considered so far all contain a role as the *subject*-**set** in the target. However in the core specification of XACML [13] the *subject*-**set** of a rule can be an arbitrary set of subjects. If this is the case, then in general *every* possible combination of subject, resource and action would need to be tested, rather than just a representative from each role combination.

8 Acknowledgments

Many thanks to Joey Coleman for guiding me through shell scripting, and John Fitzgerald for guiding me through VDM.

This work is part-funded by the UK EPSRC under e-Science pilot project GOLD, DIRC (the Interdisciplinary Research Collaboration in Dependability) and DSTL.

References

[1] D.J. Andrews, editor. *Information technology – Programming languages, their environments and system software interfaces – Vienna Development Method – Specification Language – Part 1: Base language*. International Organization for Standardization, December 1996. International Standard ISO/IEC 13817-1.

[2] J. C. Bicarregui, J.S. Fitzgerald, and P.A. Lindsay et. al. *Proof in VDM: A Practitioner's Guide*. Springer-Verlag, 1994.

[3] CSK. VDMTools. available from `http://www.vdmbook.com`.

[4] Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. Access control: principles and solutions. *Software - Practice and Experience*, 33:397–421, 2003.

[5] Kathi Fisler, Shriram Krishnamurthi, Leo A. Meyerovich, and Michael Carl Tschantz. Verification and change-impact analysis of access-control policies. In *ICSE '05: Proceedings of the 27th International Conference on Software Engineering*, pages 196–205, New York, NY, USA, 2005. ACM Press.

[6] J. Fitzgerald, I.J. Hayes, and A. Tarlecki, editors. *International Symposium of Formal Methods Europe. Newcastle, UK, July 2005*, volume 3582 of *LNCS*. Springer-Verlag, 2005.

[7] John Fitzgerald and Peter Gorm Larsen. *Modelling Systems: Practical Tools and Techniques in Software Developement*. Cambridge University Press, 1998.

[8] D. Guelev, M. Ryan, and P. Schobbens. Model-checking Access Control Policies. In *ISC'04: Proceedings of the Seventh International Security Conference*, volume 3225 of *LNCS*, pages 219–230. Springer, 2004.

[9] Graham Hughes and Tevfik Bultan. Automated Verification of Access Control Policies. Technical Report 2004-22, University of California, Santa Barbara, 2004.

[10] D. Jackson. *Micromodels of software: Modelling and analysis with Alloy*. `http://sdg.lcs.mit.edu/alloy/reference-manual.pdf`.

[11] Cliff B. Jones. *Systematic Software Developement using VDM*. International Series in Computer Science. Prentice-Hall, 1990.

[12] OASIS. Core and heirarchical role based access control (RBAC) profile of XACML v2.0. Technical report, OASIS, Feb 2005.

[13] OASIS. eXtensible Access Control Markup Language (XACML) version 2.0. Technical report, OASIS, Feb 2005.

[14] The GOLD project. `http://gigamesh.ncl.ac.uk/`.

[15] N. Zhang, M. Ryan, and D. Guelev. Evaluating access control policies through model checking. In J. Zhou, J. Lopez, R.H. Deng, and F. Bao, editors, *Eighth Information Security Conference (ISC'05)*, volume 3650 of *LNCS*, pages 446–460. Springer-Verlag, 2005.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

RSSM: A Rough Sets based Service Matchmaking Algorithm

Bin Yu and Maozhen Li

Electronic and Computer Engineering, School of Engineering and Design
Brunel University, Uxbridge, UB8 3PH, UK
{Bin.Yu, Maozhen.Li}@brunel.ac.uk

Abstract

The past few years have seen the Grid is evolving as a service-oriented computing infrastructure. It is envisioned that various resources in a future Grid environment will be exposed as services. Service discovery becomes an issue of vital importance for utilising Grid facilities. This paper presents RSSM, a Rough Sets based service matchmaking algorithm for service discovery that can deal with uncertainty of service properties when matching service advertisements with service requests. The evaluation results show that the RSSM algorithm is more effective in service discovery compared with other mechanisms such as UDDI and OWL-S.

1. Introduction

The past few years have seen the Grid [1, 2] is evolving as a service-oriented computing infrastructure. Open Grid Services Architecture (OGSA) [3], a service-oriented architecture promoted by the Global Grid Forum (GGF) [23], has facilitated the evolution. It is expected that Web Services Resource Framework (WSRF) [4] will be acting as an enabling technology to drive this further. It is envisioned that a future Grid environment may therefore host a large number of services exposing resources such as applications, software libraries, CPUs, disk storage, network links, instruments and visualisation devices. Service discovery becomes an issue of vital importance for utilising Grid facilities.

UDDI [5] has been proposed as an industry standard for Web service publication and discovery. However, the search mechanism supported by UDDI is limited to keyword matches and does not support any inference based on the taxonomies. To enhance service discovery, UDDI has been extended in such a way that service descriptions are attached with metadata, notably UDDIe [6] and UDDI-M^T [7]. UDDIe extends UDDI with user-defined properties such as service leasing, service access cost, performance characteristics, or usage index associated with a service. UDDI-M^T extends UDDI in a more flexible way allowing metadata to be attached to various entities associated with a service. The work of UDDI-M^T has been utilised in the myGrid project [8] to facilitate Grid service discovery with semantic descriptions [9]. There is also current work in implementing a service registry based on extensions to UDDI, called GRIMOIRES [10].

With the development of Semantic Web [11], services can be annotated with metadata for enhancement of service discovery. One key enabling technology to facilitate service annotation and matching is OWL-S [12], an OWL [13] based ontology for encoding properties of Web services. OWL-S ontology defines a service profile for encoding a service description, a service model for specifying the behavior of a service, and a service grounding for how to invoke the service. Typically, a service discovery process involves a matching between the profile of a service advertisement and the profile of a service request using domain ontologies described in OWL. The service profile not only describes the functional properties of a service such as its inputs, outputs, pre-conditions, and effects (IOPEs), but also non-functional features including service name, service category, and aspects related to the quality of a service. Paolucci et al. present an algorithm for matching OWL-S services [14]. This work has been extended in various way for service discovery, e.g. Jaeger et al. [15] introduce “contravariance” in matching inputs and outputs between service advertisements and service requests using OWL-S, Li et al. [16] introduce a “intersection” relationship between a service advertisement and a service request, Majithia et al. [17] introduce reputation metrics in matching services.

Besides OWL-S, another prominent effort to enhance service description and discovery is WSMO

[18], which is built on four key concepts – ontologies, standard WSDL based Web services, goals, and mediators. WSMO stresses the role of a mediator in order to support interoperation between Web services. WSMO introduces mediators aiming to support distinct ontologies employed by service requests and service advertisements.

The above-mentioned methods facilitate service discovery in some way. However, when matching service advertisements with service requests, these methods assume that service advertisements and service requests use consistent properties to describe relevant services. This is not always true for a large-scale Grid system where service publishers and requestors may use their pre-defined properties to describe services. Some properties used in a service advertisement may not be used by a service request when searching for services. Therefore, one challenging work in service discovery is that service matchmaking should be able to deal with uncertainty of service properties when matching service advertisements with service requests.

In this paper, we present RSSM, a Rough Sets [19] based service matchmaking algorithm for service discovery that can deal with uncertainty of service properties. Experiment results show that the algorithm is more effective for service matchmaking than UDDI and OWL-S mechanisms.

The remainder of this paper is organised as follows. Section 2 presents in depth the design of the RSSM matchmaking algorithm. Section 3 compares the RSSM with UDDI and OWL-S mechanisms in terms of effectiveness and overhead in service discovery. Section 4 concludes this paper and presents future work.

2. Algorithm Design

RSSM considers input and output properties individually when matching services. For the simplicity of expression, input and output properties used in a service request are generally referred to as service request properties. The same goes to service advertisements.

Figure 1 shows RSSM components for service matchmaking. The Irrelevant Property Reduction component takes a service request as an input (step 1), and then it accesses a set of advertised domain services (step 2) to remove irrelevant service properties using the domain ontology (step 3). Reduced properties will be marked in the set of advertised domain services (step 4). Once invoked (step 5), the Dependent Property Reduction component accesses the advertised domain services (step 6) to discover and reduce dispensable properties which will be marked in advertised domain services (step 7). Invoked by the Dependent Property Reduction component (step 8), the Service Matching and Ranking component accesses the advertised domain services for service matching and ranking (step 9), and finally it produces a list of matched services (step 10).

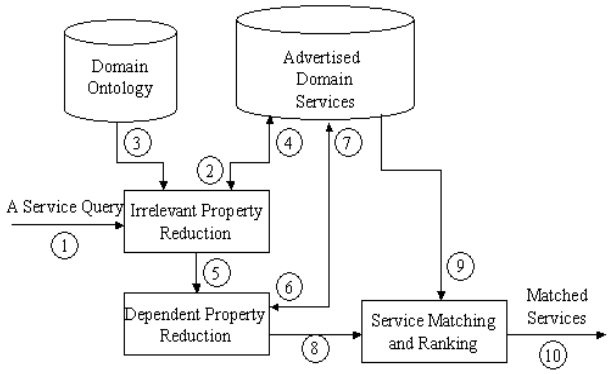


Figure 1. RSSM components.

In the following sections, we describe in depth the design of RSSM components. Firstly, we introduce Rough Sets for service discovery.

2.1 Service Discovery with Rough Sets

Rough sets theory is a mathematic tool for knowledge discovery in databases. It is based on the concept of an upper and a lower approximation of a set as shown in Figure 2. For a given set X, the yellow grids (lighter shading) represent its upper approximation, and the green grids (darker shading) represent its lower approximation.

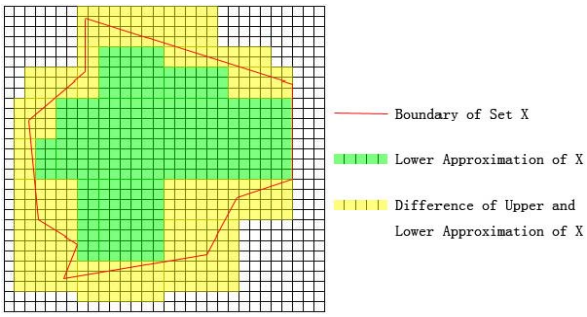


Figure 2. Approximation in rough sets.

Let

- Ω be a domain ontology.

- U be a set of N advertised domain services, $U = \{s_1, s_2, \ldots, s_N\}$, $N \geq 1$.
- P be a set of K properties used to describe the N advertised services of the set U, $P = \{p_1, p_2, \ldots, p_K\}, K \geq 2$.
- P_A be a set of M advertised service properties relevant to a service request R in terms of the domain ontology Ω, $P_A = \{p_{A1}, p_{A2}, \ldots, p_{AM}\}, P_A \subseteq P, M \geq 1$.
- X be a set of advertised services relevant to the service request R, $X \subseteq U$.
- $\underline{X}$ be the lower approximation of the set X.
- $\overline{X}$ be the upper approximation of the set X.

According to the Rough Sets theory, we have

$$\underline{X} = \{x \in U : [x]_{P_A} \subseteq X\} \quad (1)$$

$$\overline{X} = \{x \in U : [x]_{P_A} \bigcap X \neq \varnothing\} \quad (2)$$

For a property $p \in P_A$, we have

- $\forall x \in \underline{X}$, x definitely has property p.
- $\forall x \in \overline{X}$, x possibly has property p.
- $\forall x \in U - \overline{X}$, x absolutely does not have property p.

The use of "definitely", "possibly" and "absolutely" are used to encode properties that cannot be specified in a more exact way. This is a significant addition to existing work, where discovery of services needs to be encoded in a precise way, making it difficult to find services which have an approximate match to a query.

2.2 Reducing Irrelevant Properties

When searching for a service, a service requestor may use some properties irrelevant to the properties used by a service publisher in terms of a domain ontology. These irrelevant properties used by advertised services should be removed before the service matchmaking process is carried out.

Let

- p_R be a property for a service request.
- p_A be a property for a service advertisement.

Following the work proposed by Paolucci et al. [14], we define the following relationships between p_R and p_A:

- **exact** match, p_R and p_A are equivalent or p_R is a subclass of p_A.
- **plug-in** match, p_A subsumes p_R.
- **subsume** match, p_R subsumes p_A.
- **nomatch**, no subsumption between p_R and p_A.

Algorithm 1. Reducing irrelevant properties from advertised services.

```
1: for each property p_A used in advertised domain services
2:   for all properties used in a service request
3:     if p_A is nomatch with any p_R
4:       then p_A is marked with nomatch
5:     end if
6:   end for
7: end for
8: remove all p_A that are marked with nomatch
```

For each property of a service request, we use the reduction algorithm as shown in Algorithm 1 to remove irrelevant properties from advertised services. For those properties in advertised services that have a nomatch result they will be treated as irrelevant properties. Advertised services are organised as service records in a database. Properties are organised in such a way that each uses one column to ensure the correctness in the following reduction of dependent properties. As a property used by one advertised service might not be used by another advertised service, some properties may have empty values. A property with an empty value in an advertised service record becomes an uncertain property in terms of a service request. If a property in an advertised service record is marked as nomatch, the column associated with the property will be marked as nomatch. As a result, all properties within the column including uncertain properties (i.e. properties with empty values) will not be considered in service matchmaking.

2.3 Reducing Dependent Properties

Property reduction is an import concept in Rough Sets. Properties used by an advertised service may have dependencies. Dependent properties are indecisive properties that are dispensable in matching advertised services.

Let

- Ω, U, P, P_A be defined as in Section 2.1.
- P_A^D be a set of L_D properties of which each is an individual decisive property for identifying advertised services relevant to the service request R in terms of Ω, $P_A^D = \{p_{A1}^D, p_{A2}^D, \ldots, p_{AL_D}^D\}, P_A^D \subseteq P_A$, $L_D \geq 1$.
- P_A^{IND} be a set of L_{IND} properties of which each is an individual indecisive property for identifying advertised services relevant to the service request R in terms of Ω, $P_A^{IND} = \{p_{A1}^{IND}, p_{A2}^{IND}, \ldots, p_{AL_{IND}}^{IND}\}$, $P_A^{IND} \subseteq P_A, L_{IND} \geq 1$.
- $P_A^{IND\_Core}$ be a core subset of P_A^{IND} that has the maximum number of individual indecisive properties and the group of these properties in $P_A^{IND\_Core}$ are indecisive in identifying advertised services relevant to the service request R in terms of Ω, $P_A^{IND\_Core} \subseteq P_A^{IND}$.
- $IND(P_A^{IND})$ be an equivalence relation also called indiscernibility relation on U.
- f be a decision function discerning an advertised service with properties.

Then

$$IND(P_A^{IND}) = \{(x,y) \in U : \forall p_{Ai}^{IND} \in P_A^{IND}, f(x, p_{Ai}^{IND}) = f(y, p_{Ai}^{IND})\} \quad (3)$$

$$P_A^D = P_A - P_A^{IND} \quad (4)$$

The Dependent Property Reduction component uses the algorithm as shown in Algorithm 2 to find the decisive properties used by advertised services.

Algorithm 2 uses the advertised services with the maximum number of nonempty property values as targets to find indecisive properties. The targeted services can still be uniquely identified without using these indecisive properties. All possible combinations of individual indecisive properties will be checked with an aim to remove the maximum indecisive properties which may include uncertain properties whose values are empty. In the mean time, the following service discovery process can be speeded up because of the reduction of properties in matching advertised services.

2.4 Service Matching and Ranking

The Service Matching and Ranking component uses the decisive properties received from the Dependent Property Reduction component to match and rank advertised services in terms of a service request.

Algorithm 2. Reducing dependent properties from advertised services.

```
S is a set of advertised services with the maximum number of nonempty
   property values relevant to a service request;
P_A is a set of properties used by the S set of advertised services;
P_A^D is a set of decisive properties, P_A^D ⊆ P_A;
P_A^IND is a set of individual indecisive properties, P_A^IND ⊆ P_A ;
P_A^IND_Core is a set of combined indecisive properties,
   P_A^IND_Core ⊆ P_A^IND;
P_A^D = Ø; P_A^IND = Ø; P_A^IND_Core = Ø;
1:  for each property p∈ P_A
2:    if p is an indecisive property for identifying the S set of services
3:      then
4:          add p into P_A^IND;
5:          P_A^IND_Core = Ø;
6:          add p into P_A^IND_Core;
7:    end if
8:  end for
9:  for i=2 to sizeof(P_A^IND)-1
10:   calculate all possible i combinations of the properties in P_A^IND;
11:   if any combined i properties are indecisive properties for identifying
            the S set of services
12:     then
13:       P_A^IND_Core = Ø;
14:       add the i properties into P_A^IND_Core;
15:       continue;
16:     else if any combined i properties are decisive properties
17:       then break;
18:   end if
19: end for
20: P_A^D = P_A-P_A^IND_Core;
21: return P_A^D;
```

Let

- Ω, U, P, P_A be defined as in Section 2.1.
- P_R be a set of M properties used in a service request R. $P_R = \{p_{R1}, p_{R2}, \ldots, p_{RM}\}, M \geq 1$.
- P_A^D be a set of L_D decisive properties for identifying advertised services relevant to the service request R in terms of Ω, $P_A^D = \{p_{A1}^D, p_{A2}^D, \ldots, p_{AL_D}^D\}$, $L_D \geq 1$.

- $m(p_{Ri}, p_{Aj})$ be a match degree between a requested service property P_{Ri} and an advertised service property P_{Aj} in terms of Ω, $p_{Ri} \in P_R, 1 \le i \le M$, $p_{Aj} \in P_A^D, 1 \le j \le L_D$.
- $v(p_{Aj})$ be a value of the property p_{Aj}, $p_{Aj} \in P_A^D, 1 \le j \le L_D$.
- $S(R,s)$ be a similarity degree between an advertised service s and the service request R, $s \in U$.

Algorithm 3. Rules for calculating match degrees between requested service properties and advertised service properties.

```
1:  for each property pAj ∈ PA^D, v(pAj) ≠ NULL
2:    for each property pRi ∈ PR
3:      if pAj is an exact match with pRi
4:        then m(pRi, pAj) = 1;
5:      else if pAj is a plug-in match with pRi
6:          then if pRi is the kth subclass of pAj and 2≤k≤5
7:              then m(pRi, pAj) = 1-(k-1)×0.1;
8:          else if pRi is the kth subclass of pAj and k>5
9:              then m(pRi, pAj) = 0.5;
10:         end if
11:     else if pAj is a subsume match with pRi
12:       then  if pAj is the kth subclass of pRi and 1≤k≤3
13:             then m(pRi, pAj) = 0.8-(k-1)×0.1;
14:         else if pAj is the kth subclass of pRi and k>3
15:             then m(pRi, pAj) = 0.5;
16:         end if
17:     end if
18:   end for
19: end for
20: for each property pAj ∈ PA^D, v(pAj) = NULL
21:   for each property pRi ∈ PR
22:     m(pRi, pAj) = 0.5;
23:   end for
24: end for
```

Algorithm 3 shows the rules for calculating a match degree between a requested service property and an advertised service property. A decisive property with an empty value has a match degree of 50% when matching all the properties used by a service request. An advertised service property will be given a match degree of 100% if it has an exact match relationship with a property used by a service request. An advertised service property will be given a match degree of 50% if it has a plug-in relationship with a service request property and the relationship is out of five generations. Similarly, an advertised service property will be given a match degree of 50% if it has a subsume relationship with a service request property and the relationship is out of three generations.

Each decisive property used for identifying advertised services has a maximum match degree when matching all the properties used in a service request. $S(R,s)$ can be calculated using formula (5).

$$S(R,s) = \sum_{j=1}^{L_D} \sum_{i=1}^{M} \max(m(p_{Ri}, p_{Aj})) \Big/ L_D \quad (5)$$

Using the formula (5), each advertised service has a similarity degree in terms of a service request.

3. Experimental Results

We have implemented the RSSM algorithm using Java on a Pentium IIII 2.6G with 512M RAM running Red Hat Fedora Linux 3. jUDDI [20] and mySQL [21] were used to build a UDDI registry. We designed *Pizza* services for the tests using the *Pizza* ontology defined by http://www.co-ode.org/ontologies/pizza/pizza_20041007.owl. Figure 3 shows the Pizza ontology structure. The approach adopted here can be applied to other domains – where a specific ontology can be specified. The use of service properties needs to be related to a particular application-specific ontology.

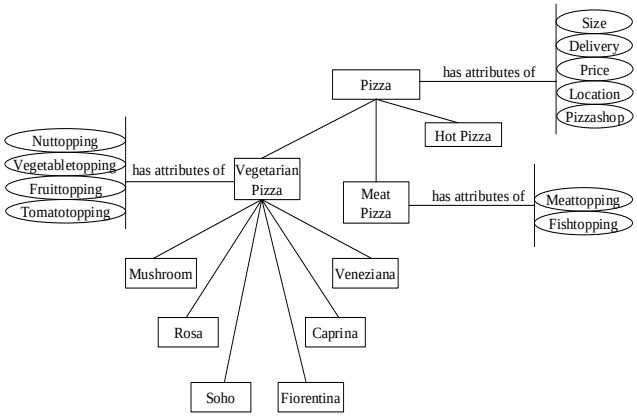


Figure 3: Pizza ontology structure.

In the following sections, we evaluate the RSSM algorithm in terms of its effectiveness and overhead in service discovery. We compare RSSM matchmaking with UDDI and the OWL-S matchmaking algorithm proposed by Paolucci et al. [14] respectively. RACER

[22] was used by the OWL-S algorithm to reason the relationship between an advertised service property and a requested service property. We implemented a simple reasoning component in RSSM to reduce the high overhead incurred by RACER.

3.1 Evaluating Effectiveness in Service Discovery

We have performed four groups of tests to evaluate the effectiveness of RSSM in service discovery. Each group had some targeted services to be discovered. Properties such as *Size*, *Price*, *Nuttoping*, *Vegetariantopping*, and *Fruittopping* were used by the advertised services. Table 1 shows the evaluation results.

Table 1. The results of effectiveness evaluation.

Service Property Certainty Rate	UDDI Keyword Matchmaking		OWL-S Matchmaking		RSSM Matchmaking	
	Correct Matching Rate	Matched Service Records	Correct Matching Rate	Matched Service Records	Correct Matching Rate	Matched Service Records
100%	50%	4	100%	3	100%	3
70%	33.3%	3	0%	0	100%	1
50%	0%	1	0%	0	100%	1
30%	0%	1	0%	0	100%	1

In the tests conducted for group one, both the OWL-S matchmaking and RSSM matchmaking have a correct match rate of 100%. This is because all services in this group do not have uncertain properties (i.e. properties with empty values). UDDI discovers four services, but only two services are correct because of the use of keyword matching (in this case making use of a categoryBag).

In the tests of the last three groups where advertised, services have uncertain properties, the OWL-S matchmaking cannot discover any services producing a matching rate of 0%. Although UDDI can still discover some services in the tests of the last three groups, the correct matching rate for each group is low. For example, in the tests of group three and group four where advertised services have only 50% and 30% certain properties respectively, UDDI cannot discover any correct services. The RSSM matchmaking is more effective than UDDI and the OWL-S matchmaking in tolerating uncertain properties when matching services. For example, the RSSM matchmaking is still able to produce a correct match rate of 100% in the tests of the last three groups.

3.2 Evaluating Overhead in Service Discovery

We have registered 10,000 *Pizza* service records in a database for testing the overhead of the RSSM matchmaking. We compare the overhead of the RSSM matchmaking with that of UDDI and the OWL-S matchmaking respectively in service matching as shown in Figure 4. We also compare their performance in accessing service records as shown in Figure 5.

From Figure 4 we can see that UDDI has the least overhead in service matching services. This is because UDDI only supports keyword matching. It does not support the inference of the relationships between requested service properties and advertised service properties which is a time consuming process.

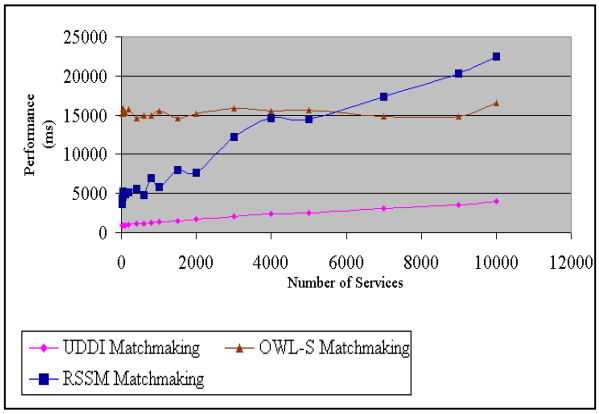


Figure 4. Overhead evaluation in matching services.

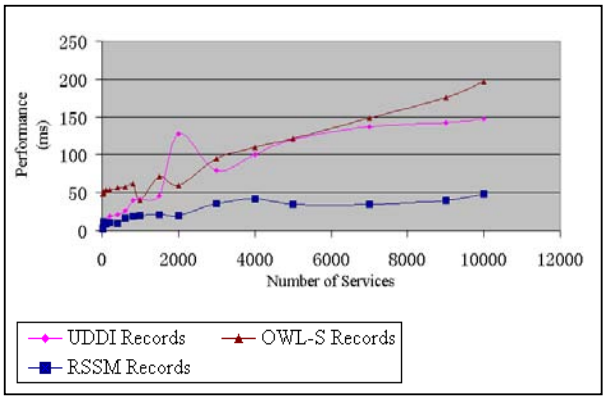


Figure 5: Overhead evaluation in accessing service records.

We also observe that the RSSM matchmaking has a better performance than the OWL-S matchmaking when the number of advertised services is within 5500. This is because the RSSM matchmaking used a simpler reasoning component than RACER, which was used by

the OWL-S matchmaking. However, the overhead of the RSSM matchmaking increases when the number of services gets larger, due to the reduction of dependent properties. The major overhead of the OWL-S matchmaking is caused by RACER, which is sensitive to the number of properties used by advertised services instead of the number of services.

From Figure 5 we can see that the RSSM matchmaking has the best performance in accessing service records due to its reduction of dependent properties. The OWL-S matchmaking has a similar performance to UDDI in this process.

4. Conclusions and Future Work

In this paper we have presented RSSM, a Rough Sets based algorithm for service matchmaking. By dynamically reducing irrelevant and dependent properties in terms of a service request, the RSSM algorithm can tolerate uncertain properties in service discovery. Furthermore, RSSM uses a lower approximation and an upper approximation of a set to dynamically determine the number of discovered services which may maximally satisfy user requests. Experimental results have shown that the RSSM algorithm is effective in service matchmaking.

As the reduction of dependent properties is a time consuming process, the future work will be focused on the design of heuristics for reducing dependent properties to speed up the process.

References

[1] I. Foster and C. Kesselman, The Grid, Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers Inc., San Francisco, USA, 1998.

[2] M. Li and M.A.Baker, The Grid: Core Technologies, Wiley, 2005.

[3] Open Grid Services Architecture (OGSA), http://www.globus.org/ogsa/

[4] Web Services Resource Framework (WSRF), http://www.globus.org/wsrf/

[5] Universal Description, Discovery and Integration (UDDI), http://www.uddi.org/

[6] A. ShaikhAli, O. F. Rana, R. J. Al-Ali, D. W. Walker: UDDIe: An Extended Registry for Web Service. Proceedings of SAINT Workshops, Orlando, Florida, USA, 2003.

[7] S. Miles, J. Papay, V. Dialani, M. Luck, K. Decker, T. Payne, and L. Moreau. Personalised Grid Service Discovery. IEE Proceedings Software: Special Issue on Performance Engineering, 150(4):252-256, August 2003.

[8] myGrid project, http://www.mygrid.org.uk

[9] S. Miles, J. Papay, T. Payne, K. Decker, and L. Moreau. Towards a protocol for the attachment of semantic descriptions to grid services. Proceedings of the Second European across Grids Conference, volume 3165 of Lecture Notes in Computer Science, Nicosia, Cyprus, January 2004.

[10] L. Moreau, Grid RegIstry with Metadata Oriented Interface: Robustness, Efficiency, Security (GRIMOIRES). http://www.grimoires.org/, 2005.

[11] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. Scientific American, Vol. 284 (4), pages 34-43, 2001.

[12] OWL Services Coalition, OWL-S: Semantic Markup for Web Services, white paper, Nov. 2003; www.daml.org/services/owl-s/1.0.

[13] S. Bechhofer, F. Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language Reference. W3C Recommendation, Feb. 2004.

[14] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic matching of web service capabilities. Proceedings of 1st International Semantic Web Conference. (ISWC2002), Berlin, 2002.

[15] M. C. Jaeger, G. Rojec-Goldmann, G. Mühl, C. Liebetruth, and K. Geihs. Ranked Matching for Service Descriptions using OWL-S. Proceedings of KiVS 2005, Kaiserslautern, Germany, Feb. 2005.

[16] L. Li and I. Horrocks. A software framework for matchmaking based on semantic web technology. Int. J. of Electronic Commerce, 8(4):39-60, 2004.

[17] S. Majithia, A. S. Ali, O. F. Rana, D. W. Walker: Reputation-Based Semantic Service Discovery. Proceedings of WETICE 2004, Italy.

[18] Web Service Modeling Ontology (WSMO), http://www.wsmo.org

[19] Z. Pawlak. Rough sets. International Journal of Computer and Information Science, 11(5):341-356, 1982.

[20] jUDDI, http://ws.apache.org/juddi/

[21] mySQL, http://www.myswl.com

[22] V. Haarslev and R. Möller. Description of the RACER System and its Applications. Proceedings International Workshop on Description Logics (DL-2001), Stanford, USA.

[23] Global Grid Forum (GGF), http://www.ggf.org

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Semantic Units for Scientific Data Exchange

Kieron R Taylor, Ed Zaluska, Jeremy G Frey*
University of Southampton, School of Chemistry,
Highfield, Southampton, SO17 1BJ, UK
j.g.frey@soton.ac.uk

Abstract

All practical scientific research and development relies inherently on a well-understood framework of quantities and units. While at first sight it appears that potential issues should by now be well understood, closer investigation reveals that there are still many potential pitfalls. There are a number of well-publicised case studies where lack of attention to units by both humans and machines have resulted in significant problems. The same potential problems exist for all e-Science applications whenever data is exchanged between different systems; an unambiguous definition of the units is essential to data exchange. E-science applications must be able to import and export scientific data accurately and without any necessity for human interaction. This paper discusses the progress we have made in establishing such a capability and demonstrates it with prototype software and a small but varied library of units.

1 Introduction

All practical scientific research and development relies inherently on a well-understood framework of quantities and units. We use the term quantity here as used in the Green Book[1] to describe concepts of length, time and so on, rather than the more common "dimension". An essential part of all scientific training is directed to the study and understanding of this topic. While at first sight it appears that potential issues should by now be well-understood, closer investigation reveals that there are still many potential pitfalls. There are a number of well-publicised case studies where human misinterpretation of unit information has resulted in significant problems.

Exactly the same potential problems exist in all e-Science applications whenever data is exchanged between different systems - an unambiguous definition of the units is essential. While significant progress has been made using XML[2] (and XML derivatives) to describe such data, these solutions only address the issue of markup, and not interoperability. Such an approach falls well short of the functionality necessary for a Semantic Grid[3,4]. e-Science applications in a Semantic Grid must be able to import and export scientific data (and in fact any numerical data) accurately and without any necessity for human interaction. Without a system to decide whether it is receiving quantities with the correct units for its use, nothing but careful data entry prevents error or disaster.

This paper discusses the progress we have made in establishing such a capability. We conclude that RDF[5] provides a suitable and practical means to describe scientific units to enable their communication and interconversion, and thereby solve a major problem in reliable exchange of scientific data.

2 Existing unit systems

We might choose to store all numbers according to SI conventions[1] and therefore contain ourselves within a single system of units and quantities. This is a workable solution in many situations. Unfortunately many fields of science have units that predate SI and are conveniently scaled for analysis, and so storing these values in SI units would be inconvenient as well as requiring careful conversion. If on the other hand we decide to retain the original units we face the prospect of other people choosing to use different units. This is almost inevitable given that different purposes may lead us to use any of SI, British Imperial, US Imperial ("English units"), CGS, *esu*, *emu*, Gaussian and atomic systems or even more ancient and country-specific systems. It is quite common to have two measurements on different scales that might well be the same, and yet we have no way of comparing the two without the aid of a pocket calculator and appropriate numerical constants.

To understand fully the problems of describing units it is useful to examine what a measurement consists of: A numerical value with units. These two components must not be-

come separated or the value loses all meaning. The following demonstrates how a measure of solution strength can be deconstructed:

Solution strength of 0.02 mol dm^{-3}

0.02 The value of the measurement.

mol One unit, the mole indicating one Avogadro's constant of molecules.

dm^{-3} A second unit - "per decimeter cubed". This may be further decomposed into:

deci An SI prefix for 1/10th.

meter The SI unit for length.

$^{-3}$ An exponent of the preceding unit, present if not equal to one.

Confining ourselves to the SI system for the moment, we note the possibility of multiple units for one measurement, and that each unit may be preceded by a scaling factor such as milli or mega. In addition to this each unit may be raised to a power, either positive or negative, typically with a magnitude of six or less.

3 Conversion between unit systems

The great breadth of scientific and engineering endeavour over many centuries has led to a wide variety of systems with a legacy of conveniently sized units selected for a particular purpose. Inevitably units from different systems are encountered in the same setting and conversion must take place so that values may be compared or combined. The ramifications of imperfect conversions are considerable, as NASA discovered to their cost when their Mars Climate Orbiter was destroyed in 1999[6] due in part to the incorrect units of force given to the software. In 1983 the "Gimli Glider" incident[7] involving a Boeing 767 running out of fuel occurred because of invalid conversion of fuel weights. These two very high profile events are extreme examples of a problem that is encountered all the time. Lack of precision and rigour lead to costly mistakes in something that is not difficult but requires attention.

The majority of conversions are exchanges of one unit for another, such as celsius for kelvin, or yards for meters. Such conversions present no challenge, but not all conversions are so straightforward. Knowledge of the quantities (or dimensions) a unit relates to is vital to deciding what conversions are meaningful. For example, the knot is a recognised measure of speed in common use. It is a compound unit and must be separated into the quantities of length and time in order to compare it with other measures of speed. By the same token, the watt is an approved SI measure of power but it is also common for data sources to report the same quantity expressed in joules per second. How can computer software equate these two concepts?

Various efforts have been made in computer science to maintain and convert units alongside their values in software with mixed results. To do so within Object Oriented systems requires a certain inventiveness in order to "treat a single entity as both a type and a value"[8]. These coding complexities can be avoided by leaving the computational logic the same and separating the units handling into a separate software layer which is what we provide for here.

A more complicated issue is exemplified by older measurements of pressure based on a column of mercury. A column height in millimeters of mercury has been a long-established method of monitoring atmospheric pressure, but of course this is not a pressure at all, rather a length. If treated as a length for conversion purposes we may only convert from millimeters or inches to some other length of a mercury column. While this is entirely reasonable, it is not particularly useful. Some form of "bridge" is required to make the transition from a length to a pressure but it is entirely dependent on the material. This is more an issue of physical science and not obviously within the scope of units description so we treat it as a secondary concern.

Something definitely within the scope of units description but equally contentious are ratios. Treatment of ratios has been hotly debated by standards committees over the years. With all ratios such as LD50s (a lethal dose that kills 50% of a test group) and the concentration of drug in formulations, one must keep the division between unit and measurement clear. Although the units of a ratio cancel and it becomes a unitless value, the value has no meaning without context. LD50s are performed on all manner of small organisms, and it may be important to note that the dose was issued in grams per kilogram of body mass of the chosen organism. One cannot for example convert such a ratio of weights into other units if the units have been cancelled out. Some of this information is relevant to unit description, while the remainder falls into the domain

of experimental description and needs further consideration

It is clear that any system to aid unit conversion must capture the units themselves, but also the quantities to which the units apply. Unfortunately the issue of quantities is also complicated by differences in unit systems. The *esu* and *emu* systems were created to simplify the mathematics of electromagnetism and operate in only three dimensions rather than the four required to handle the same information under SI. While it is possible for *esu* and *emu* based values to work in a four-dimensional system they are commonly applied in three such that there is no need for charge in *esu*, or current in *emu*. This "mathematical trickery" makes values from these systems dimensionally different to other systems and demands non-trivial transformations to switch between them. Most other unit systems are dimensionally consistent with SI and hence can be addressed all at once. Special consideration is needed to for work involving electromagnetism.

4 Machine-readable unit descriptions

Having established the need for and complexities in describing units of measure, we come to the issue of the technology to use. XML and its derivatives address the problem of data exchange by making all data self-describing. By eschewing proprietary data formats, we make it easier for software authors to work with the data. Consequently parsing of data can be a much more robust process.

Several organisations are in the process of creating systems to make units transferable, including NIST's unitsML[9], the units sections within GML[10] and SWEET[11] ontologies from the Open Geospatial Consortium and NASA respectively. None of these systems has yet been finalised and may yet be improved upon. At this time there are no complete computerised units definitions endorsed by any standards organisation, and this means that there is no accepted procedure to express scientific units in electronic form. Wherever the problem arises, people have either created their own systems that can cope with their immediate needs or resorted to plain text, thereby condemning their data to digital obscurity.

Unit conversion and having the concepts of units within software are not particularly new ideas, as illustrated by the many conversion tools presently available and a vast amount of work on ontologies expressed in the language of LISP such as the EngMath ontology[12]. Indeed, ontologies of bewildering complexity exist to describe many things but there is little evidence of them being applied to real problems. As Vega *et. al.* explained[13], knowledge sharing has a long way to go in order for these ontologies to be reused, let alone spread beyond their original setting of knowledge engineering and artificial intelligence.

The XML schemas developed thus far tend to propose that one should have a definition for mol dm^{-3} and miles per hour as single entities, and within that definition are descriptions of what units and powers compose it is composed of. This makes description straightforward, as shown by the GML fragments below.

```
<measurement>
 <value>10</value>
 <gml:unit#mph/>
</measurement>

<DerivedUnit gml:id="mph">
 <name>miles per hour</name>
 <quantityType>speed</quantityType>
 <catalogSymbol>mph</catalogSymbol>
 <derivationUnitTerm uom="#mile" exponent="1"/>
 <derivationUnitTerm uom="#h" exponent="-1"/>
</DerivedUnit>
```

One potential problem with this approach is in the vast numbers of permutations possible to accommodate the many different ways people use units in practice. There are perhaps ten different prefixes in common use in science, so at least in principle we may have ten versions of each unit, and with compound units it might be common to talk of moles, millimoles or micromoles per decimeter, centimeter or meter cubed. We would then have around one hundred useful permutations and many more possibilities.

Clearly such a system is more useful if it considers non-SI units such as inches, pounds and so on. Every combination of two units together results in another definition leading to a finite but practically endless list of definitions. This is exemplified by the units section of the SWEET ontology. SWEET presently addresses a relatively small set of units around the basic SI units, and already the list is many pages of definitions with a very precise syntax required to invoke them. In a long list, humans will have difficulty locating the correct entities and both processing and validating the schema becomes increasingly difficult. There is already considerable scope for typographical errors when writing programs to use ontologies, and the bigger and more complex the ontology the greater the problem becomes.

A more tractable but neglected alternative to the above approach is to explode the units when they are invoked as follows:

```
<measurement>
 <value>10</value>
 <unit>
  <unitname>mile</unitname>
  <power>1</power>
 </unit>
 <unit>
  <unitname>hour</unitname>
  <power>-1</power>
 </unit>
</measurement>
```

or in more condensed form

```
<measurement>
 <value>10</value>
 <unit id=#mile power="1"/>
 <unit id=#hour power="-1"/>
</measurement>
```

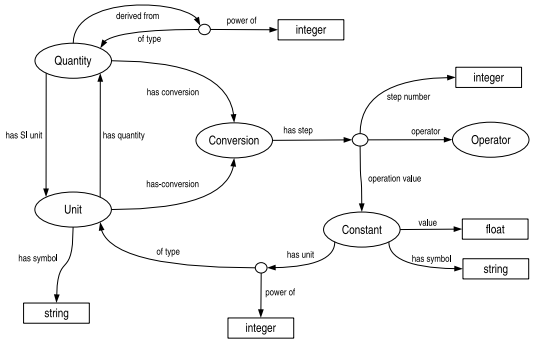
Clearly this approach requires more data to describe the units for each measurement, but it does dramatically reduce the size of the dictionary required to interpret it. The cornucopia of distinct combinations are reduced down to a succinct construct of two units and a definition of a prefix.

5 The proposed units schema

We have elected to use RDF to describe both unit information in documents and to describe the relationships between units. This more specialised form of XML is convenient on account of our existing RDF knowledgebase, but can also be readily embedded in web documents. All RDF statements join pieces of information together, and general RDF interpreters know how to operate on this data. Conversions between units are entirely based on rules such as what a mile may be converted into, and constants such as what must be used to convert a mile into the equivalent length expressed in meters. If we were to store this information in XML just as GML does, we must interpret the XML into these rules and essentially repeat the work that RDF already covers. The web compatibility of RDF allows unit relationships and definitions to be exchanged across the internet in the same way as the data itself. This is a very important factor when considering standardisation of unit magnitudes.

There is a more subtle detail favouring RDF over XML to describe units and their conversions. XML is fundamentally a hierarchical tree-like structure with children, parents and siblings. This is not the case with RDF, which allows more complex networks to be formed. Much more powerful data description is possible without being limited to single level relationships such as sibling or parent. We can also add concepts such as "similar to" and branch across trees of the data. This is valuable in the context of units owing to the complex web of relationships between units and quantities. The limitations on a schema are very much down to what is logically sensible and reasonable to program. It is possible to make non-hierarchical networks computationally intractable so such flexibility should only be employed with care.

Figure 1: Units schema visualisation



The schema we propose to handle units is depicted in figure 1. Nodes represent RDF resources (subjects or objects) and arcs represent predicates. Values in rectangles are literal values and may be subjected to XML data types. The labels for nodes are defined as follows:

Quantity A description of the type of a unit, also sometimes called dimension. SI base quantities include mass, length and time, while derived quantities include force, energy and velocity.

Unit Scientific units describing exactly what "one of these" is measured in. Units can be SI or from other measurement systems such as Imperial.

Conversion An anonymous entity grouping together all steps of a conversion process.

Operator A mathematical operator, such as multiply or divide.

Constant The constant of a conversion including both a value and units.

The construct begins with the unit class. Subclasses of units from particular unit systems also exist. Meter, second and yard are

examples of instances of the unit class, and not a class in itself as is commonly encouraged in ontology creation. This is a case of the frequently encountered class/instance dilemma. In principle there is only one true measure for a quantity in a given unit system. In this case we defer to SI to endorse one standard value for the magnitude of one meter, one second etc. The difference between a US gallon and a UK gallon is just one example of many semantic collisions that require clear description and differentiation.

Each unit instance is linked to its corresponding quantity (length, time, mass). A derived quantity (such as volume or velocity) may be constructed from other quantities in which case "derived-from" links imported from the ontology instruct the system what other quantities can be substituted for the derived quantity and to what order. Quantities all have a standard SI recommended unit, and where they are derived, they have a connection to the base quantities from which they are derived. This enables consistency checking between units by using their quantities regardless of whether the quantity is base or derived.

The conversions themselves are expressed as a series of computational operations, each consisting of both scalar values and units. The combination of units and values collected as one constant make it possible to perform conversions without prior knowledge of the outcome. Although having units on the constants may seem unnecessary, it supplies additional rigour to the conversion process, as well as lucidity to the conversion itself. An instruction might be to multiply by 3600 seconds per hour, cancelling existing units and reminding us what that scalar transformation represents. It also allows support for conversions that connect different quantities by fundamental physical relationships, as discussed later. This arrangement makes it possible to infer the units that result from a conversion rather than having to specify it in the ontology yielding great benefits in managing the units library and simplifying implementation.

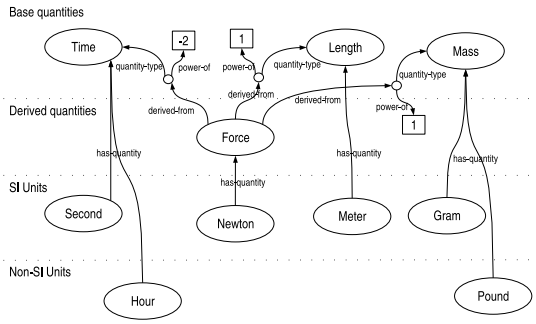
We investigated the possibility of storing conversions in some form of equation, but this demanded either a complete development of an equation system or the use of an existing system such as MathML[14] or the less commonly known OpenMath[15]. This solution proved far more complicated than practical and was discarded in favour of a stepwise system that still retains the content and reversibility of an equation. As long as the reversibility is retained, conversions need not be described in both directions, thereby simplifying the library even further.

In order to maintain the integrity of the units library, a number of rules must be observed that may be enforced with an ontology.

- Every unit must relate to a base or derived quantity.
- Quantities that are not one of the 7 SI base quantities must be derived from a combination of those 7 quantities.
- All non-SI units must have a conversion to SI base units or combinations of units.
- Quantities may have conversions which alter the dimensionality of the system using SI units for the conversion.

The above system is complex and creates an extensive network of units and quantities containing many cross-links. This is expected and cannot be simplified any further without compromising the function of the system. A part of the library is illustrated in figure 2, showing the conceptual separation of base quantities, derived quantities and the units that correspond to them.

Figure 2: Instances of units and quantities



The units are unscaled and without exponents in order to avoid the combinatorial issue discussed earlier. SI prefixes such as milli and mega are defined and invoked as separate entities. Supplementary information such as preferred abbreviations are also defined as required. Non-SI units have only one conversion to the SI equivalent and no others. This helps to minimise the number of conversions required and avoids issues of multiple redundant paths to the same result. If we wish to convert between two measurements in the Imperial system, a route is found via the SI unit, retaining any exponents. For example: Miles $\Longrightarrow$ Meters $\Longrightarrow$ Feet instead of Miles $\Longrightarrow$ Feet. The only caveat to this is computational precision, as floating point arithmetic in a binary

computer inevitably introduces small errors. This can be countered by use of appropriate algebraic mathematical libraries, and this needs to be considered when software is written to handle chains of conversions.

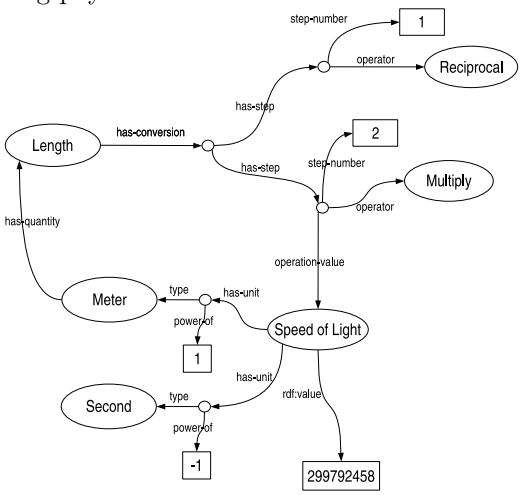
6 Physical Equivalence Conversions

Yet another set of conversions exist in science that are extremely useful but completely transform both the value and the quantity of a measurement. Any physicist will know that mass can be translated into energy with the correct fundamental constant. Likewise, spectroscopists regularly convert wavelengths (or inverse wavelengths, specifically the wavenumber in cm^{-1}) into energies, typically in electron volts, or joules. These transitions from one set of base units to another are made possible by equations which use fundamental constants incorporating units of their own. While far from obvious and not a pure unit conversion, these scientific equivalences are useful and having such conversions automated is even more helpful than just providing normal conversions. To that end we have used the same conversion description method for quantity-quantity conversions as well as unit-unit conversions. The mathematical processes implied by equations such as $E = h\nu$ and $E = mc^2$ can be described in exactly the same way as the process that translates from miles to meters. The only differences are their attachment to quantities rather than units, and the logic required to decide when to use them. This is illustrated in figure 3, where the conversion stems from the length quantity. Unfortunately it is not obvious how to limit the use of this conversion to values that do not relate to electromagnetic radiation. This is an issue of context involving both purpose and meaning that goes beyond the scope of units and conversions. This wider context of describing the measurements themselves might require a separate ontology to embrace all of science and engineering.

7 Test implementation

A program has been written ("uniterator", available from the authors on request) that reads in the ontology each time a conversion is required, and accepts RDF files containing values and units along with a request to convert to another set of units. The process followed by the program is outlined in figure 4. The program reduces the input units to SI base units by expanding any derived SI quantities, and performing all necessary conversions to SI base units. The same process is applied to the requested units, performing conversions in reverse. The simplified request and starting units are compared for quantity and unit consistency, i.e. that the request has asked for a reasonable operation such as length to length, and nothing like length to volume. At this point, an inconsistency in quantities leads to a systematic exploration of possible quantity-quantity conversions, such that useful equalities for frequencies and energies can be included amongst other physical equivalences. All combinations of up to an arbitrary limit of three consecutive quantity-quantity conversions are considered, and the appropriate conversions performed if a quantity match can be achieved. If the quantities are deemed compatible and the conversion is a success, the result has already been computed and is written out in an RDF wrapper. Otherwise the request is rejected as meaningless or beyond the scope of the program.

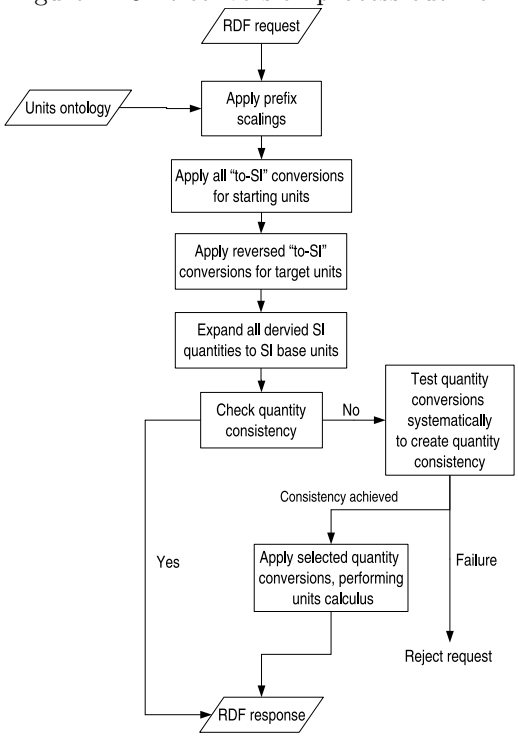
Figure 3: Describing a wavelength equivalence using physical constants



The "uniterator" has been tested with the following successful conversions using a relatively limited ontology of units:

- 10.5 mJ $\longrightarrow$ 0.0105 W
- 10 mg dm^{-3} $\longrightarrow$ 4.55e-02 kg $gallon^{-1}$
- 10 Fahrenheit $\longrightarrow$ -12.22 Celsius
- 10 Fahrenheit $\longrightarrow$ 0 m Prevented due to incompatible quantities
- 5 lbs $inch^{-2}$ $\longrightarrow$ 3515.35 kg m^{-2}
- 735 nm $\longrightarrow$ 4.08e+14 s^{-1}

Figure 4: Unit conversion process outline



- 30 knots ⟶ 34.52 miles per hour
- 6 GHz ⟶ 3.98e-21 mJ
- 200 cm^{-1} ⟶ 3.97e-21 J

The whole process relies heavily on the commutative nature of the conversion processes. This may lead to problems with conversions involving a translation of origin, such as with the Celsius temperature scale, but this can be resolved with a more rigorous program. Since this program is a proof-of-concept script, it will not be developed further to ensure absolute reliability. At present it is capable of performing conversions involving simple temperatures on outmoded scales, but may fail with particular combinations of units.

It should be noted that it was necessary to use grams as the base SI unit instead of kilograms. Although incorrect as far as SI is concerned, it allows complete divorcing of prefix and unit. The outputs of this software can be refined to present data according to SI recommendations and is not a significant problem. Some additional care is needed in encoding of conversions that normally rely in some way on the kilogram, such as measures of energy.

An RDF or XML request for conversion takes the following form:

```
<ch:Quantity>
 <ch:has-value>10</ch:has-value>
 <has-unit>
  <Unit rdf:type="#Fahrenheit">
   <power-of>1</power-of>
  </Unit>
 </has-unit>
 <has-desired-unit>
  <Unit rdf:type="#Celsius">
   <power-of>1</power-of>
  </Unit>
 </has-desired-unit>
</ch:Quantity>
```

The program returns a response in the same format containing both new units and the value.

8 Conclusions

In summary, previous attempts have been made to define units for computer software, but all of them have run into difficulties at various stages of their development. Successfully designing a system that solves all of the possible problems has proven to be very challenging because of the endless variations of unit application and the tendency for people to define their own units. The problem is simply too broad for any one person to have experience of all units and this has led to systems which are unintentionally incapable of tackling some units satisfactorily. The boundaries between unit and measurement are somewhat blurred and this clouds the issue further.

The units system outlined here has been developed with a heavy emphasis on facilitating implementation and usefulness. It provides a manageable way to make scientific units machine-parseable and interconvertible on the semantic web. RDF is used to create a network of units and quantities that can be effortlessly extended with new units and conversions without requiring any rewritten software. It provides several advantages over existing XML methods by controlling the ways in which units relate to each other, and by clearly addressing issues of dimensionality, convenience and functionality. A design decision has been made to keep the central dictionary and relationships as small and elegant as possible while retaining scope for even the most exotic of conversions between systems with the same number of dimensions. The specialised systems used for electromagnetism remain a problem to be addressed in future work. Although not addressed here, It is entirely reasonable to have a parallel ontology for the *esu* and *emu* systems with appropriate conversions. Such a process involves many intricacies and only applies to a relatively specific area of science hence we have not yet attempted to resolve it. The defini-

tions of quantities and units in our system will almost certainly make such a transformation manageable.

Another key factor that is not provided is any intelligence. The system is not rich enough to identify misuses of units, and cannot hope to address some of the finer points such as restrictions on when Hz may be used instead of s^{-1} and when a second relates to an angle. Attempting to address these more subtle distinctions could easily lead to an ontology far too complicated for useful implementation. It is perhaps more suitable for this issue to be addressed at the interface level rather than in the underlying data. The key to successful deployment is to design a system to be as universally understandable as possible. Once a proven and complete system is agreed upon, a more expressive ontology language such as OWL can be used to restrict and validate units and conversions more comprehensively.

An area neglected by this paper is that of uncertainty. Strictly speaking, no measurement is complete without a declaration of precision. This conspicuous absence is for a variety of reasons. Firstly, expressions of error and precision come in many forms, relative and absolute, all of which must be accounted for. Secondly there are many ways to mark up precision, and we do not presume to force an approach on the reader. The units system presented here is intended to demonstrate what can be done and to raise awareness of the requirements for machine-readable units. The issue of measurement mark up including precision, units and domain relevance (to prevent spurious conversions) is deserving of a much lengthier discussion. There is no reason why these features cannot be added to our schema or software.

We have demonstrated that the semantic technology RDF can provide a practical method to describe and communicate scientific units necessary for complete description of quantities, together with methods for comparison and conversion of those units. The system provides a basis for an ontology that will enable the automated validation of the nature of a quantity, its compatibility with the units and its comparability with other quantities to provide the necessary and appropriate conversions between unit systems.

References

[1] I. Mills, T. Cvitas, K. Homann, N. Kallay, and K. Kuchitsu, *IUPAC Quantities, Units and Symbols in Physical Chemistry*, Blackwell Science, 2 ed., 1993.

[2] World Wide Web Consortium, Extensible markup language http://www.w3.org/XML/, viewed 2005.

[3] D. De Roure, N. Jennings, and N. Shadbolt Research agenda for the semantic grid: A future e-science infrastructure Technical Report UKeS-2002-02, National e-Science Centre, December , (2001).

[4] D. De Roure, N. Jennings, and N. Shadbolt In *Proceedings of the IEEE*, pages 669–681, 2005.

[5] World Wide Web Consortium, Resource description framework http://www.w3.org/rdf/, viewed 2005.

[6] NASA, Mars climate orbiter believed to be lost http://mars.jpl.nasa.gov/msp98/orbiter/, 1999.

[7] M. Williams, *Flight Safety Australia*, July-August (2003).

[8] E. E. Allen, D. Chase, V. Luchangco, J.-W. Maessen, and G. L. S. Jr. In J. M. Vlissides and D. C. Schmidt, Eds., *OOPSLA*, pages 384–403. ACM, 2004.

[9] National Institute of Standards and Technology, Units markup language http://unitsml.nist.gov/, 2003.

[10] Open Geospatial Consortium, GML - the geography markup language http://www.opengis.net/gml/, viewed 2005.

[11] R. Raskin, M. J. Pan, I. Tkatcheva, and C. Mattmann, Semantic web for earth and environmental terminology http://sweet.jpl.nasa.gov/index.html, 2004.

[12] T. R. Gruber and G. R. Olsen In J. Doyle, P. Torasso, and E. Sandewall, Eds., *Fourth International Conference on Principles of Knowledge Representation and Reasoning*, 1994.

[13] J. C. A. Vega, A. Gomez-Perez, A. L. Tello, and H. S. A. N. P. Pinto, *Lecture Notes in Computer Science*, **1607**, 725 (1999).

[14] W3C Math working group, Mathml 2.0 http://www.w3.org/Math/, 2001.

[15] OpenMath Society, Openmath http://www.openmath.org/, 2001.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

The ISIS Facilities Ontology and OntoMaintainer

Louisa Casely-Hayford, Shoaib Sufi

CCLRC e-Science Centre, CCLRC Daresbury Laboratory
Warrington WA4 4AD,UK

Abstract

This paper presents an ISIS facilities ontology based on keywords in the ISIS Metadata catalog (ICAT) and OntoMaintainer, a web application to facilitate the collaborative development of ontologies. The ISIS facilities ontology aims to organize and make keywords in the ISIS Metadata Catalogue more explicit which will improve the search and navigation of data by category. The Ontology Maintainer allows users to view current versions of the ontology, and send feedback on concepts modeled back to the maintainers of the ontology. To ensure ease of use, this service has been made available to the user community via the World Wide Web.

1. Introduction

ISIS, the worlds leading pulsed neutron and muon source, is one of the major large scale facilities operated by CCLRC. The ISIS Metadata Catalogue (ICAT) is a twenty year back catalogue of experiments conducted at ISIS. Currently the ISIS facility produces about 700GB of combined Neutron and Muon data each year, and the addition of a second target station will expand ISIS to twice its current size, hence the volume of data generated is set to rise [1]. Consequently, the efficient storage, retrieval and management of data is vital for the full value of these data resources to be realised [2]. To address this problem, e-science is developing numerous software solutions and ontologies are seen one of these useful approaches. Ontologies provide central controlled vocabularies that can be integrated into catalogues, databases, web publications and knowledge management applications. They are becoming increasingly popular amongst members of the scientific community, because they offer a powerful means to formally express the nature of a domain.

At present over 10,000 keywords describing experiments are housed in ICAT many of which are synonyms. These keywords are used to index experimental studies, however this is seen as a limited method as these free text keywords have no context, much implicit meaning and are hard to map by non-experts to terms used by facilities in the same domain and harder still to those outside. For example, a particular keyword 'HRPD' designates an instrument which is a 'powder diffractometer'; there are situations in which other collaborating Neutron facilities (e.g. SNS at ORNL in the US) understand the meaning of a powder diffractometer but do not understand that the cryptic 'HRPD' refers to such an instrument. The creation of ontologies aids in the mapping of concrete manifestations of familiar terms in one domain as well as related concepts in different domains. Thus the creation of such ontologies at CCLRC could aid in the cross facility search of related scientific data from the various science facilities housed at CCLRC e.g. ISIS, Central Laser Facility (CLF) and Diamond Light Source (DLS). An ISIS facility ontology was built using the web ontology language (OWL) within the java based editing environment Protégé, to address the need for the keywords in ICAT to be organized, classified and formally defined.

Ontologies are not static and continually evolve, therefore with time they will need to be updated and extended. A single ontology is usually built co-operatively by a group of people in different geographical locations. This is because the knowledge contained within an ontology represents a common view shared within a community. The construction and maintenance of an ontology is a difficult task, and dialogue is essential to reach a consensus and distribute information. This is a very important activity within communal design, however there are few tools that address this problem. For these reasons the Ontology Maintainer was developed, as it aims to facilitate the collective task of designing, building, extending and updating ontologies. This paper describes the creation of the ISIS Facilities ontology and Ontology Maintainer. In the following sections, several aspects of ontologies, ontology building, and the structure of the ISIS facilities ontology and the OntoMaintainer will be examined. At the end of the paper, some directions for future work in this area will be mentioned.

2. Background

2.1 What is an Ontology?

The word ontology stems from Philosophy, where it means a branch of metaphysics that investigates and explains the nature, essential properties and relations of all beings [2]. To date there are several existing definitions of the word ontology. One of the most quoted definitions in literature by the ontology community is by Tom Gruber. He defined an ontology as "an explicit specification of a conceptualization" [3]. In 1997 Borst slightly altered Gruber's definition saying that; "Ontologies are defined as a formal, specification of a shared conceptualization" [4].
In Gruber's definition 'Conceptualization' refers to an abstract model of some phenomena in the world by having identified the relevant concepts of those phenomena. 'Explicit' means that the type of concepts used, and the constraints on their use are explicitly defined. 'Formal' indicates that the ontology should be machine readable, which excludes natural language. 'Shared' reflects the belief that an ontology captures consensual knowledge meaning that it is not private to some individual, but accepted by a group [5].

In all, ontologies are explicit specifications of the concepts in a given field, and of the relationships between those concepts [6]. The required minimum for an ontology is to include concepts, definitions of concepts, and defined relationships between the concepts. By defining shared and common domain theories, ontologies help both people and machines to communicate concisely, which promotes knowledge reuse and integration [3]. Ontologies are widely used for different purposes (natural language processing, e-commerce, knowledge management, intelligent information integration and the semantic web) and by different communities (knowledge engineering, bioinformatics, databases and software engineering) [7].
An example of an ontology widely used by biologists to search for associations of genes and gene products is GO. The Gene Ontology (GO) project is a collaborative effort to address the need for consistent descriptions of gene products in different databases. The backbone of GO is made up of three ontologies, that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner [8].

2.2 Building an Ontology

Ontology building is more of a craft than an engineering process. Several research groups propose various methods more commonly known as methodologies for building ontologies. There is no consensus between these groups and each employs its own methodology. During the process of building an ontology several questions arise. Some of these include:

- What domain will the ontology cover?
- What will the ontology be used for?
- What types of questions should the information in the ontology provide answers for (competency questions)?

The construction of ontologies is necessarily an iterative process, and answers to these questions may change during the life cycle of the ontology. The four basic aspects to consider when creating an ontology are content (the content of the ontology), the language in which it is implemented, the methodology which has been followed to develop it, and the software tools used to build and edit the ontology. Furthermore the type of language and development environment chosen is dependent on the type of ontology being built. It is important to know what the ontology is going to be used for, and how detailed the ontology should be, so that the possibility of over modelling (e.g. attempting to model the "whole world") is lessened.

2.3 Ontology Languages

A formal language is used to encode the ontology and different languages provide different facilities [9]. The most recent development in standard ontology languages is the web ontology language (OWL) from the World Wide Web Consortium (W3C) [10]. OWL is a language based on description logics, and can be used explicitly to represent the meaning of terms in vocabularies, and the relationships between those terms [10]. The Web Ontology language (OWL), is a revision of DAML+OIL web language and has more facilities for expressing meaning and semantics than XML, RDF and RDF-S.

OWL ontologies consist of Individuals, Properties and Classes or Concepts (terms relevant to the domain). Classes (concepts) are a concrete representation of concepts, and in OWL classes are interpreted as sets that contain individuals [9]. A concept can be anything about which something can be said, for example human, John or pet. Classes within the ontology

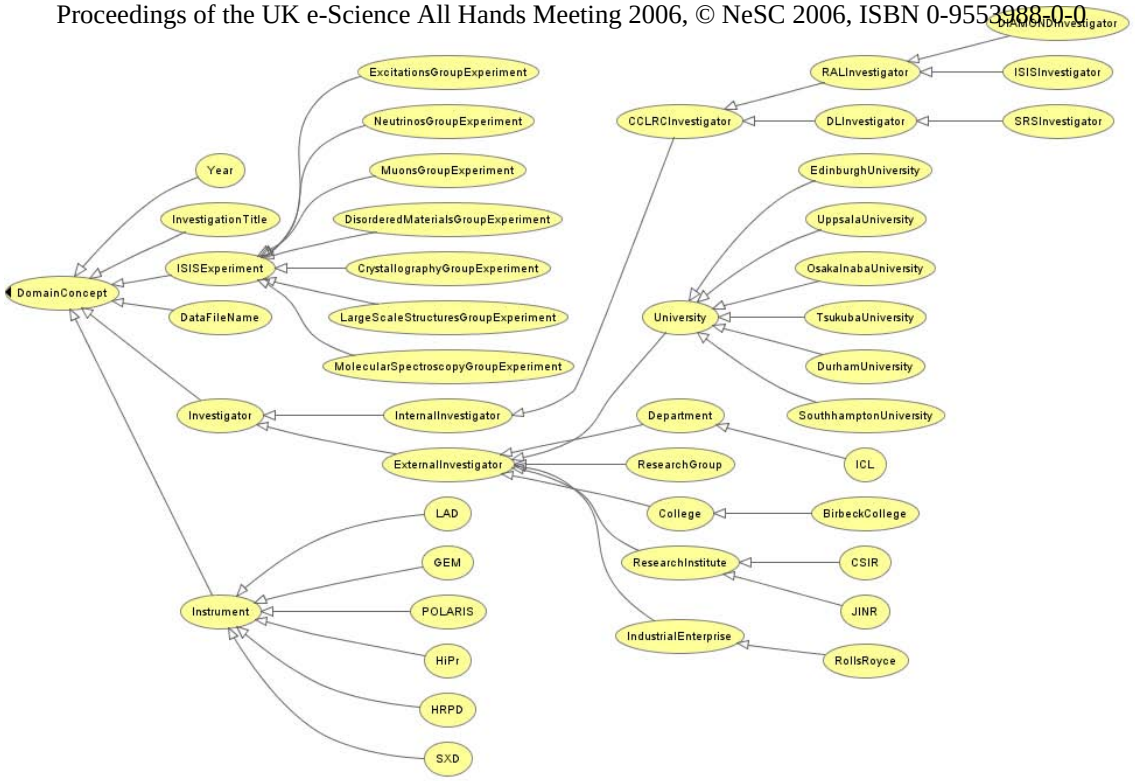


Figure 1 Diagram of Overall Hierarchy of ISIS Facilities Ontology

are normally organised in taxonomies with a subsumption ("subclass") hierarchy. OWL classes can be specified as logical combinations (intersections, unions, or complements) of other classes, or as enumerations of specified objects. OWL can also declare properties, organize these properties into a "sub-property" hierarchy, and provide domains and ranges for these properties [10]. Properties allow general facts about the members of classes and specific facts about individuals to be asserted and also represent relationships between two individuals. For example the property hasSibling might link the individual Matthew to the individual Gemma. There are two main types of properties, object properties and data type properties. Data type properties are relations between instances of classes and RDF literals and XML Schema data types. Object properties are relations between instances of two classes. Properties may have a domain and a range specified. The domains of OWL properties are OWL classes, and ranges can be either classes or externally-defined datatypes such as string or integer. Properties link individuals from the domain to individuals from the range. Individuals, represent objects in our domain of interest [10]. For example Italy, England China and France are individuals in the class Country.

OWL provides three sublanguages, namely OWL Lite, OWL DL and OWL Full [10]. OWL Lite supports those users primarily needing a classification hierarchy and simple constraints. OWL DL supports those users who want the maximum expressiveness while retaining computational completeness for reasoning systems. OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees [10]. OWL's logical model allows the use of a reasoner which can check whether or not all of the statements and definitions in the ontology are mutually consistent, and can also recognise which concepts fit under which definitions. The reasoner can therefore help to maintain the hierarchy correctly. This is particularly useful when dealing with cases where classes can have more than one parent [11].

2.4 Ontology Editing Environments

Ontology development or engineering tools include suites and environments that can be used to build a new ontology from scratch or by reusing existing ontologies [12]. Since the mid-nineties there has been an exponential increase in the development of technological platforms related with ontologies [7]. Protégé is one of the most widely used ontology building tools and has been developed by the Stanford Medical Informatics (SMI) Group at Stanford University [13]. Protégé instances, slots and classes roughly corresponds to OWL Individuals, Properties and Classes (concepts).

3. ISIS Facilities Ontology

As mentioned earlier, an ISIS facilities Ontology was built to compile a comprehensive structured vocabulary of terms representing the keywords describing experiments in ICAT. This ontology will be used as a means of grouping data across studies, disambiguating keywords and improving the search and navigation of data by category. The Protégé OWL plug-in was selected to create the ISIS facilities ontology. The web ontology language (OWL) was chosen because of its ability to provide formal semantics, built-in reasoning support and additional features such as metadata annotation and ontology versioning. OWL DL was used as it supports maximum expressiveness while allowing the use of a reasoner [1]. After close examination of the keywords and their definitions, it was decided that they would be grouped into five categories namely: datafile, instrument, investigation_title, investigator and year. This is because all of the keywords in ICAT fell into these five groups.

Examples of keywords in these five categories are:

1. HRP00145.RAW which is a datafile name.
2. HRPD which is a High Resolution Powder Diffractometer one of the many instruments used in experiments at the ISIS facility
3. Hydrazinium which is an investigation title, chemical names and compounds were used as investigation titles of experiments in ICAT
4. 1986 which is the year in which a particular experiment was conducted
5. JINR (Joint Institute for Nuclear Research in Russia) which is the name of an investigator.

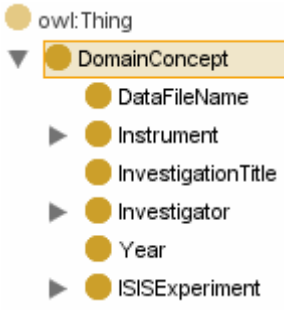


Figure 2 Snapshot of general ontology hierarchy in Protégé

The field of ontological engineering is still young and is not as developed as that of knowledge engineering. Although there is some cooperative experience of building and using ontologies, up to now there are no standardised methodologies for building ontologies [14]. One of the most recognized ontology construction guidelines, developed by Thomas Gruber, was followed to build the ISIS facility ontology [15]. A top-down modelling approach was used to build the ontology with an overview of the system being formulated first, afterwhich each part of the system was then refined by designing it in more detail. The overall hierarchy of the ontology can be viewed in Figure 1. The classes DataFileName, Instrument, InvestigationTitle, Investigator and Year formed the core of the ontology (Figure 2). A class ISISExperiment was added to represent Experiments carried out at ISIS by different scientific groups (Figure 4). More subclasses were added to Instrument, ISISExperiment and Investigator to represent the different kinds of investigator, instrument and ISIS experiments (Figure 1).

All ISIS Instruments (e.g. HRPD, POLARIS, LOQ, IRIS Figure 1) were added as subclasses of the class Instrument to define the different kinds of instruments used in experiments at ISIS (Figure 3). A subsumption relationship exists between classes and subclasses in OWL, therefore for example the intstrument HRPD (the High Resolution Powder Diffractometer) is a subclass of the superclass Instrument. This subsumption relationship indicates that HRPD is a kind of Instrument and an Instrument is HRPD. Therefore if M is an instance of class HRPD, and HRPD is a subclass of Instrument, then we can infer that M is an instance of Instrument (Figure 3).

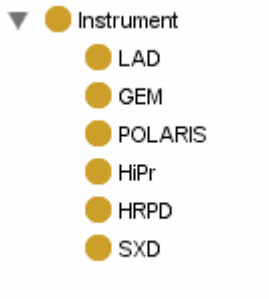


Figure 3 Snapshot of subclass Hierarchy of Instrument Class

Subclasses representing experiments carried out by different scientific groups (e.g. CrystallographyGroupExperiment), at ISIS were added to the class ISISExperiment (Figure 4). This is to facilitate common instrument requirements related to different research groups and to provide a focus for user interaction.

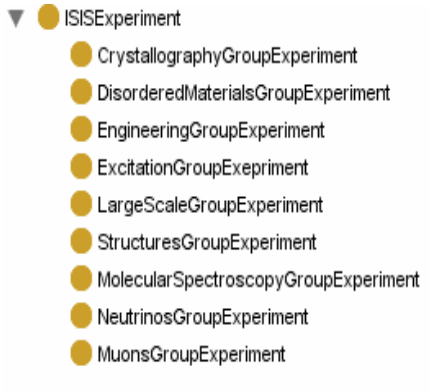


Figure 4 Snapshot of subclass Hierarchy of ISIS Experiment Class

The Class Investigator was subclassified into internal and external investigator. Next external investigators were subclassed into College, Department, ReaserchGroup, ReasearchInstitute and University (Figure 5). Furthermore internal investigators were subclasified into DL (Daresbury Laboratory) and RAL (Rutherford Apppleton) investigators (Figure 5).

Figure 5 Snapshot of subclass Hierarchy of Investigator Class

Properties were created to provide a link or relationship between individuals in the class ISISExperiment and individuals in the other five main classes. The properties hasDataFileName, hasInvestigationTitle, hasInvestigator, hasUsed and wasConductedIn were created and domain and range constraints set in Protégé. All five properties have ISISExperiment as their domain and DataFileName, InvestigationTitle, Investigator, Instrument, and Year as their range respectively (Figure 6).

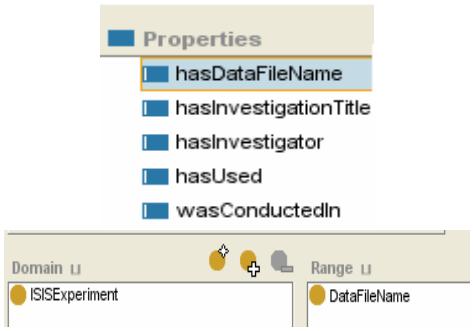


Figure 6 Snapshot of Properties in Properties Tab

4. Ontology Maintainer

Consensus on Concepts modelled in the ISIS Facilities ontology, was achieved through a series of interviews with domain experts. During the design and creation process, there was a difficulty in sharing current versions of the ontology with our collaborators at ISIS.

This is because to view the hierarchical structure of the ontology, scientists would have to download and install Protégé locally which is a time consuming and complicated process. Although there are a few web-based ontology library systems and editing environments for communal building, many of these tools are: complicated to use, not easily accessible, do not provide a user-friendly interface, and do not provide a graphical view of the hierarchy of the ontology. For these reasons the Ontology Maintainer was developed to aid the community to remotely view current versions of the ontology. It is purely a visual tool aimed at the social engineering (collective effort) aspect of ontology design and is not for the editing or building of ontologies.

The Ontology Maintainer is a web application with a client server-based architecture that was developed using Java Server Pages, servlets, CSS (Cascading Style Sheets) and Tomcat. The

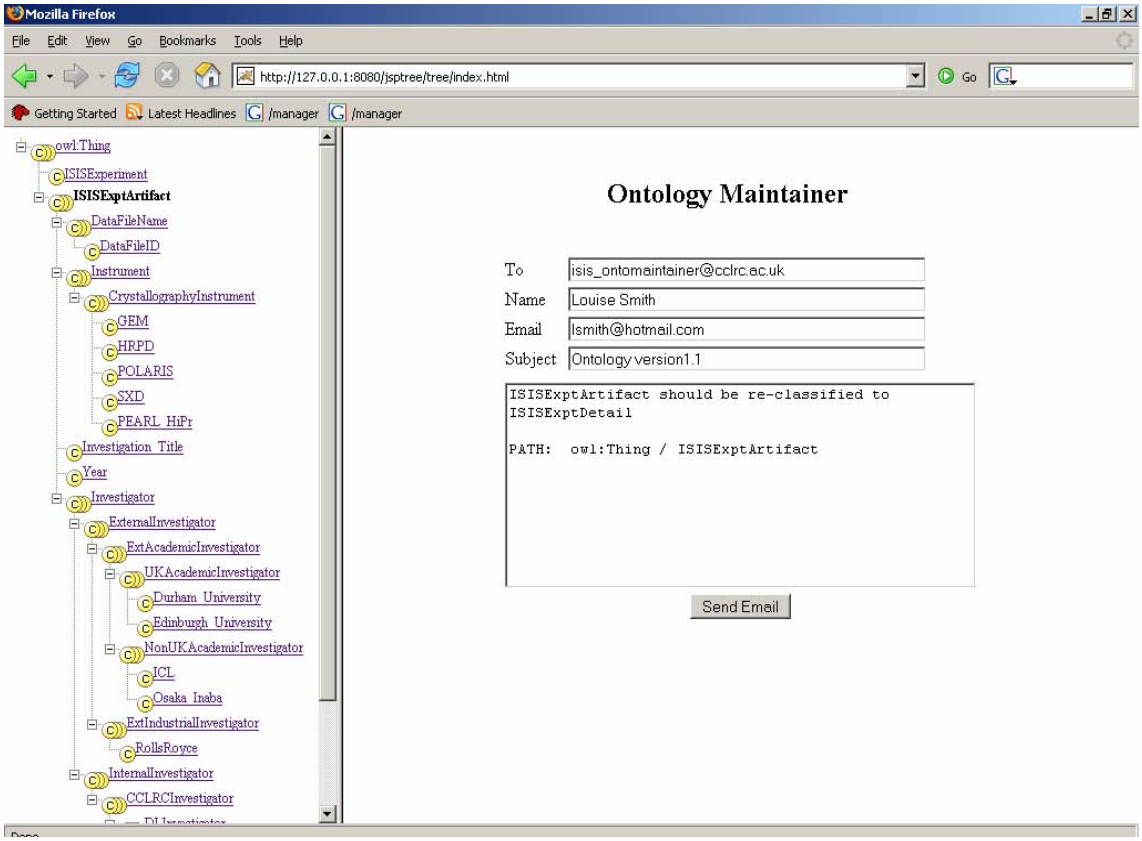


Figure 7 Screenshot of the OntoMaintainer

major advantage of this tool is that it provides an easily accessible, user friendly method of displaying the hierarchy of the ontology being built. In addition to visualisation, it enables users to submit feedback through a simple web form to maintainers of the ontology.

The web-page is divided into two panels one displaying the tree hierarchy and the other showing a web form. The first panel shows a tree hierarchy with class names displayed alongside each of the tree nodes, users can expand the tree fully by clicking on nodes or classes in the tree. The web application is modular and the look and feel of the tree can be changed easily. To send comments users can click on the class they wish to comment on in the first panel and this will generate a path of that particular class through the ontology, which is added to the text area of the comment form. Next users can enter their full name, email address, subject and comments in the second panel, as shown in Figure 7. Parameters entered in the form will be sent directly to the maintainer of the ontology.

Ontologies are constantly being changed due to technological advances, errors in earlier versions of the ontology, or the release of a novel method of modelling a domain. The process of updating an ontology is not an easy task, as there needs to be a general consensus amongst members of the community. The Ontology Maintainer will provide a huge impact on this process, as it will allow individuals to easily access current versions of the ontology through the World Wide Web.

5. Future work

The current ISIS facilities ontology serves as a base for future ontologies, and in the near future it will be expanded with more concepts, properties and restrictions. An ISIS Online Proposal System has been developed to automate the capture of metadata [1]. This system enables users of the ISIS facility to electronically create, submit and collaborate on applications for beam-time [1]. This new electronic proposal system provides access to a rich source of metadata which can be fed directly into the experimental set. Creation of

the ISIS facilities ontology has generated interest by ISIS in the use of ontologies to mark up proposals submitted through the online proposal system. Further ontologies for sample, experiment and investigator are being developed to mark up submitted proposals. These separate modular ontologies will be developed and fed back into the Metadata catalog. Once feedback is submitted and a consensus is achieved, a person (the designated Facility ontology maintainer) will update the ontology, which will automatically serve these new terms to reflect a shared understanding. Ontologies will facilitate searching of data by category and grouping of data into keywords across studies.

The ultimate aim of these ontologies is to support or promote cross-facility searching. We can envisage an ontology at each of CCLRC's large scale facilities ISIS, CLF (Central Laser Facility) and DLS (Diamond Light Source), with mappings between them such that users can search data from their perspective and receive relevant hits from other domains. The User Interface of the Ontology Maintainer will be improved through the addition of properties. Properties will be displayed in the user interface to enable relationships between individuals in classes to be shown. Additionally a graphical view of the ontology will be generated with the click of a button. Also feedback entered by users will be stored in a file system or database. Finally the tree hierarchy in the ontology maintainer will be made more dynamic through automatic updating of classes for example based on records in a database derived from the master ontology designed in an ontology engineering tool e.g. Protégé.

6. Conclusion

The ISIS facilities Ontology serves as a stepping stone to the creation of further ontologies, which will be used to help maximise the value of data collected at ISIS. Ontologies will function to improve the access, navigation and reuse of these large scale data resources. Throughout this process, collaboration between scientists at ISIS and ontology builders will be made easier through the use of the Ontology Maintainer.

7. References

[1] Damian Flannery
ISIS Data, Metadata, Proposals and the CCLRC Data Portal
http://lns00.psi.ch/nobugs2004/papers/paper00147.pdf

[2] Diana Marcela Sánchez, José María Cavero, Esperanza Marcos
On models and ontologies
http://kybele.escet.urjc.es/PHISE05/papers/sesionIV/SanchezCaveroMarcos.pdf

[3] T.R. Gruber
A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220, 1993
http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html

[4] A.J. Duineveld, R.Stoter., M.R. Weiden, B. Kenepa, and V.R. Benjamins. (1999) Wondertools? A comparative study of ontological engineering tools.
http://sern.ucalgary.ca/KSI/KAW/KAW99/papers/Duineveld1/wondertools.pdf

[5] R. Studer, V.R.Benjamins., D. Fensel (1998) Knowledge engineering: principles and methods. Data and Knowledge Engineering 25, 161–197.

[6] M. Fernaandez-Loopez, A. Goomez-Peerez, A. Pazos-Sierra, J. Pazos-Sierra, Building a chemical ontology using METHONTOLOGY and the ontology design environment, IEEE Intelligent Systems & their applications 4 (1) (1999) 37–46
http://www.aegean.gr/culturaltec/Kavakli/MIS/papers/Corcho_2003.pdf
[7] Oscar Corcho, Mariano Fernandez-Lopex, Asuncion Gomez-Perez. Data &Knowledge Engineering 46(2003) 41-64. Methodologies, tools and languages for building ontologies. Where is their meeting point?
http://www.aegean.gr/culturaltec/Kavakli/MIS/papers/Corcho_2003.pdf

[8]M Ashburner et al, Gene Ontology: tool for the unification of biology. *Nature Genetics* **25**, 25 - 29 (2000
http://www.nature.com/ng/journal/v25/n1/pdf/ng0500_25.pdf

[9] M. Horridge, H. Knublauch, A. Rector, R. Stevens, C. Wroe (2004) A Practical Guide To Building OWL Ontologies Using The. Protege-OWL Plugin and CO-ODE Tools. Edition 1.0.
http://www.coode.org/resources/tutorials/ProtegeOWLTutorial.pdf

[10] Mike Dean, Dan Connolly, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL web ontology language reference. W3C Working Draft, 31 March 2003. http://www.w3.org/TR/2003/WD-owl-ref-20030331

[11] R.Stevens, C.Goble, S. Bechhofer (2001) Ontology-based Knowledge Representation for Bioinformatics. University of Manchester http://www.cs.man.ac.uk/~stevensr/papers/briefings-ontology.pdf

[12]Lambrix P, Habbouche M, Perez M. Evaluation of ontology development tools for bioinformatics.Department of Computer and Information Science, Linkopings universitet, 581 83 Linkoping, Sweden. patla@ida.liu.se http://bioinformatics.oxfordjournals.org/cgi/reprint/19/12/1564

[13] N.F. Noy, R.W.Fergerson, M.A. Musen (2000) The knowledge model of protege-2000: combining interoperability and flexibility. In: 12th International Conference in Knowledge Engineering and Knowledge Management (EKAWÕ00), Lecture Notes in Artificial Intelligence (Springer, ed.), pp. 72, Berlin.

[14] Oscar Corcho , Mariano Fernandez-Lopez , Asuncion Gomez-Perez. Methodologies, tools and languages for building ontologies. Where is their meeting point? Facultad de Informatica, Universidad Politecnica de Madrid, Campus de Montegancedo s/n, Boadilla del Monte, Madrid 28660,Spain (2001) http://www.aegean.gr/culturaltec/Kavakli/MIS/papers/Corcho_2003.pdf

[15] Thomas R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing.Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University. http://www.cise.ufl.edu/~jhammer/classes/6930/XML-FA02/papers/gruber93ontology.pdf

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Service Composition in the Context of Grid

Xiaofeng Du, William Song, Malcolm Munro
Computer Science Department, University of Durham
DH1 3LE, Durham, UK
{xiaofeng.du, w.w.song, malcolm.munro}@dur.ac.uk

Abstract

Grid computing has become an important new research field. The goal of the Grid computing infrastructure is to pervasively access the computational resources available on the Internet. The web service technology has the potential to achieve the goal of grid computing because of its self-contained, self-describing, and loosely coupled features. The resources can be discovered and shared through the web service's process interface. In this paper we discuss how the web services can be composed from a conceptual perspective and propos a context based semantic model for better describe web service in order to realize automatic service discovery, composition, and invocation.

1. Introduction

Current researches on the World Wide Web and the Grid, as well as Web Services, have come to a consensus issue. That is, how to extract and represent semantics for the Web information and the Grid recourses. As known to us, the World Wide Web provides an infrastructure for information exchange and interoperation and the Grid provides an infrastructure for resource sharing and cooperation [2]. The interoperability and collaboration requires common understanding of information/data and resources and the understanding requires canonical and well-formed semantic description. The semantic description is used so that the human and machine can understand each other [15].

The context of application of the Web and the Grid is **e-Science**, which studies how computer and communication technology can support and enhance the scientific process by enabling scientists to generate, analyse, share and discuss their insights, experiments and results in a more effective manner [16]. The **Grid** is the underlying computer infrastructure that provides these facilities. **Web Service**, realized in Grid services, uses a standardized XML messaging system and a stack of communication protocols to makes software available over the Web [1]. One aspect of its significance lies in composition of software components available on the Grid to form an application/service for users to accomplish certain tasks. How to discover, analyse, form, and compose resources and services for users' need depends strongly on the understanding of meanings of resources/services. In other words, semantic description for the resources/services is indispensable in development of useful and powerful Web/Grid based applications. A well defined semantic description model can enable automatic service discovery, invocation, and composition.

2. Web Service, Grid Service and Grid Infrastructure

A **Web service** is a software system designed to support interoperable machine-to-machine interaction over a network that has an interface described in a machine-processable format (specifically WSDL [4]) [17]. **Grid services** are defined by the Open Grid Services Architecture (OGSA) [3] as mechanisms for creating, managing, and exchanging information among entities. A Grid service is a Web service that conforms to a set of conventions (interfaces and behaviours) that define how a client interacts with a Grid service.

In spite of the above definitions, we can put services into the following types [5]:

1. To exchange information and documents;

2. To interactively and cooperatively perform functions;

3. To share available resources.

The first type is the Web systems or software for information exchange. For the second one, we consider that web service is a chain of (sub-) services in order to accomplish a user given task. The services in the chain are loosely coupled. Based on requirements, best match or suitable match is found between two services. If our task is to calculate (a*x+b*y), then we assume that the functions * (multiply) and + (addition) are available on the Web. A web service to accomplish this task is to find the related sub-services, apply the parameters, compose the services, and return the result, using various protocols, and specifications languages [6]. Figure 1 illustrates the relationship between task and services, and how the services can be formed into a service flow to achieve the task.

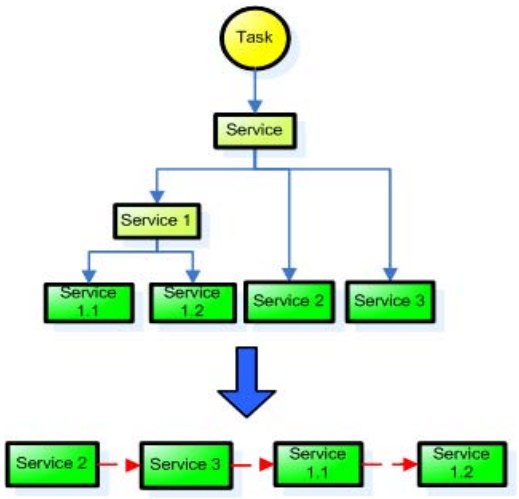


Figure 1: Service Decomposition and Service Flow

Grid service is considered to be of the third type, which emphasizes cooperation and sharing of resources (including data) among entities [2]. In particular, to share and interoperate various resources, e.g. CPU time, storage, files, from physically different nodes (entities) in a grid to form a virtual organization. For example, a virtual music album can be created through linking together music pieces from different websites or PCs.

In the grid infrastructure, the resource sharing is task driven [2]. A resource consumer issues tasks to request resource sharing from the resource providers. The web services can be considered as an interface between tasks and resources, which can consume suitable resources to achieve a certain task. In figure 2, the diagram illustrates the relationships among resource consumer, task, web service, resource, and the grid infrastructure. Under the grid infrastructure, resource consumer issues tasks; the tasks motivate the design of the web services; the web services consume resources to achieve tasks.

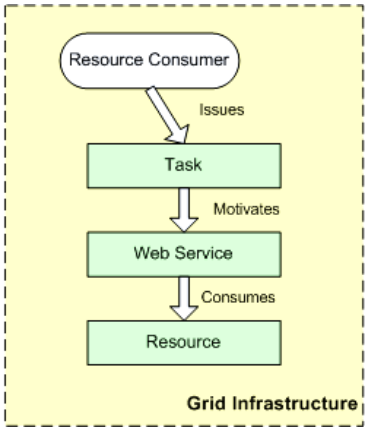


Figure 2: Relationships among task, web service, resource, and grid infrastructure

With an exponentially increasing amount of information, documents, resources, and services available on the Web, to find a good and reasonable match between the users' requirements and the capabilities of Web/Grid services as well as to form a suitable composition out of the service components to serve a demanded activity is a prominent problem. That is because we lack an effective and efficient means to describe services, components, and objects existing in the Web.

3. Conceptual Service Composition

The web services published on the Internet are mostly atomic services which only provide simple and primitive functions. Therefore, if a service provider wants to provide better services or promote more efficient sharing of resources, publishing composite services is a sound solution. Service composition can be done through identifying sub tasks, locating suitable sub-services, and formatting the sub-services into a service flow, and executing the service flow to achieve a task which is the goal of the composite service.

The difficult steps to be realized in the process of service composition are locating suitable sub-services and formatting sub-services into a service flow. Locating suitable services will be discussed in next section to see how the semantic web technology can benefit service selection. This section will focus on what aspects need to be considered when formatting the selected sub-services into a service flow by linking relevant sub-service together.

When you link two services together, in fact you are linking the former service's outputs with later service's inputs, i.e. the later service's inputs can be inputted by the value from the former service's outputs. This linking is based on the compatibility of the input and output data type, satisfiability of per and post conditions, and the similarity of the inputs and outputs' semantics. These are the basic criteria that need to be considered before joining two services together. However, for sufficiently achieving a real task, just consider the basic criteria are not enough, some other contextual information also need to be considered [7], such as:

- **Time**: When two services are running in parallel, they have to make sure they return the outputs at the same time before their outputs go into next service's inputs.

- **Data Consistency**: One service's output and another service's input have the same data type and semantic meaning does not mean the data consistency is satisfied. For example, one service's output is a weight in gram, but another one's input needs a weight in kilogram. In this example both service's input and output data type are '*double*' and both of their semantic meaning are '*weight*', but the input and output of these two services cannot be directly linked together.

- **Location**: Sometimes the location of a service also can affect the service composition. For example, one service requires an address as an input and another service can return an address, but there is a case that these two services cannot be linked together. That is when one service is in UK and another is in US.

Once a composite service has been built, the service provider needs to publish its inputs and outputs as public interfaces for the service requester to invoke. The inputs and outputs of a composite service are generated from some of its internal sub-services' inputs and outputs. If we use a directed graph ***G(V, E)*** to represent all the services in the grid environment and their possible relationships[1], then a composite service can be represented as a sub-graph of ***G***.

$$Sub\_G_1(V_1, E_1) \subseteq G$$

The ***Sub_G*₁** together with its context can be represented as another sub-graph of ***G***.

$$Sub\_G_2(V_2, E_2) \subseteq G$$

$$Sub\_G_1 \subseteq Sub\_G_2$$

Then we can represent the context of the composite service in the grid environment, which is difference between ***Sub_G*₂** and ***Sub_G*₁**

$$C(V_c, E_c) = Sub\_G_2 - Sub\_G_1$$

The set of arcs $\boldsymbol{E_c}$ in ***C*** represents the inputs and outputs of the composite service.

$$E_c = I \cup O$$

Where

I: A set of inputs of the composite service.

O: A set of outputs of the composite service.

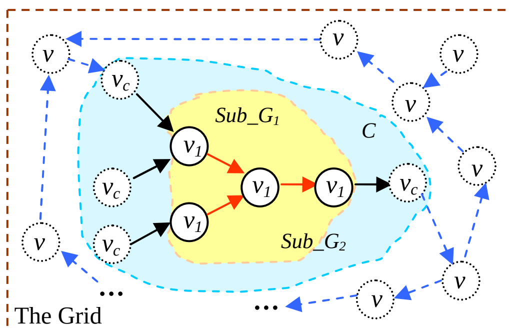


Figure 3: Composite Service, Context and Grid

Figure 3 shows a graphical illustration of the conceptual representation of the relationship among composite service, its context, and the grid environment. In the diagram, the red arrows represent the internal relationships between sub-

[1] **Note**: the arcs in the graph ***G*** are uncertain due to different composition requirements. Here just captures a possible situation.

services inside the composite service, the black arrows represent the inputs and outputs of the composite service, and the blue dashed arrows represent the relationships between the services in the grid environment.

4. Semantic Service Description Aspects and Matchmaking

As mentioned in the previous section, one of the difficult steps to be realized in the process of service composition is locating suitable sub-services. The reason is because the current web service description technology, e.g. WSDL, only provides syntactic description of a service rather than semantic description [8]. A WSDL description does not provide the information about what the service can do, thus a user has to read an extra description to get the functionalities of the service. On the other hand, the web service global registry UDDI only provide keyword matching searching rather than semantic searching, thus, it is difficult for a user to accurately find a suitable service. Therefore, integrating semantics into the current web service description and discovery technologies is crucial to improve the usability of web services and to achieve automatic service composition and resource sharing under grid infrastructure.

4.1 Semantic Service Description Aspects

A most important issue in semantic description for objects is the correct and exact capture of semantics. Lara et al. [10] and Fensel et al. [9] have discussed the semantic description requirements for describing a web service's capability, such as pre-condition, post-condition, textual description, services, and identifier. However, only integrating semantics is not sufficient to fully address a service, the context relevant information about a service cannot be ignored when describing a service. A semantic definition in a dictionary manner is not feasible because vague meaning is not only indispensable but necessary for better understanding as well. It is also difficult to provide all possible circumstances or instances of a concept (of an object), if not impossible. Actually, in most cases, people understand each other in a certain context. For example, when we talk about "jaguar" together with "BMW" and "Volvo", it is almost for certain that we mean a car instead of an animal. Using a contextual model, we can better identify the meaning of a given object and therefore select a most suitable service or information object.

Therefore, we have developed our initial contextual based semantic service description model which integrates the context together with semantics to better describe a service [7]. This model addresses a service from six aspects:

1. **IOPE**: This aspect addresses the service's input data type, output data type, pre-condition, and effects. It also addresses the semantic meaning of the inputs and outputs.

2. **Metadata**: This aspect addresses the non-functional description of the service including identifier, natural language description, location, quality attributes, and category.

3. **Ontology**: This aspect addresses the concept and domain of the service.

4. **Upper Compositionality**: This aspect addresses which kind of services can be composed by using this service. This aspect indicates the relationships between this service and other services.

5. **Lower Compositionality**: This aspect addresses what kind of services can be used to compose this service. This aspect also indicates the relationships between this service and other services.

6. **Resource**: This aspect addresses what kind of resources the service will consume. The resources include renewable resources, such as bandwidth, memory allocation, and CPU usage, and consumable resources [11], such as time. The reason for considering the resources in a service description is because the processes require resource and the processes decide how the service behaves. Therefore, the resources information can be an important criterion to identify a service. However, if a service provider thinks the resources allocation information could be private, he can hide this information from the service requesters.

Figure 4 illustrates the relationship between the six aspects for addressing a service.

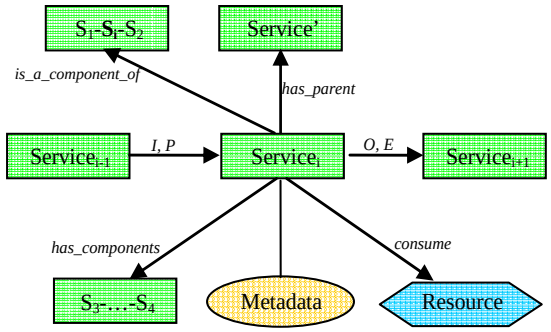


Figure 4: Contextual based Semantic Model for Service Description

The notation used in figure 4 is:

1) S_i=Service$_i$
2) **Service'** can be either the parent or ancestor of **Service$_i$**
3) **S_1**, **S_2**, **S_3**, and **S_4** are any services.
4) ***I, P*** is the Inputs and Pre-condition, and ***O, E*** is the Outputs and Effects.

4.2 Matchmaking

Based on the contextual based semantic model, we propose a semantic similarity computation approach to compare and measure the semantic similarity between a requested service and a candidate service [6]. This approach contains three steps. The first step is to compare individually the requested service (which is expanded to a sequence of services) and the candidate services from the service pool. The result is a set of ranked candidate service nodes. In the second step, we search for those candidate service pairs that meet the service pairs in the requested service sequence. The result is a set of ranked candidate service pairs. In the last step, the service pairs in the candidate service pair set are assembled into a number of sequences of services according to the requirements of the requested service sequence. In the following, we discuss in detail these three comparison and matchmaking steps.

Node comparison. To find a semantic match of two services n and m we compare their contextual characteristic sets λ(n) and λ(m) based on the contextual based semantic model. We have developed three semantic matchmaking methods for λ. In order to reflect the semantic complexity of contextual based semantic descriptions for services and domain knowledge, we also assign a set of weights to the semantic characteristic set λ(). For a given service s in the service flow, we get a set of pairs (s_k, n_k), where s_k is a service from the service pool, and n_k is an associated number to indicate how close the service s_k is semantically to s. When there are a large number of candidate services, we need to set a threshold value to contract the candidate set to a reasonable size.

Pair comparison. The second step is semantic comparison and matchmaking for service pairs. We consider two adjacent nodes, t_i-t_{i+1}, in the requested service sequence. After the first step we got two candidate service sets, S_i for t_i and S_j for t_{i+1}. Using the service composite definition, we can generate a number of service composites s_i-s_j, where $s_i \in S_i$ and $s_j \in S_j$. The services in each pair are restrained by the interdependent relationship in terms of their IOPE. Simply speaking, if there is a suitable IOPE interdependence (or match) between two service s_i and s_j and a semantic inclusion relation between t_i and s_i, and t_{i+1} and s_j, respectively, the pair s_i-s_j is a semantically compatible candidate pair matching the pair t_i-t_{i+1}. Similarly, the ranking and threshold mechanisms are applied to reduce the size of the candidate pair set.

Sequence comparison. The last step is sequence semantic matchmaking, where we create a sequence of services from the above-obtained candidate service pairs. Through iteratively applying the definitions given above, we get a number of candidate service sequences that match the requested sequence. We select the one best fit the need. Here we should emphasize the important role that the service scheduling plays in this step. When all candidate services are dynamically coupled and composed, it is possible that some sequences may contain services which suddenly become unavailable and require instantaneous replacements. In addition an important step in the service scheduling process is to quantitatively measure semantic distance between candidate services, service pairs, service sequences and given service sequence (task) and its components (such as given tasks and task pairs).

4.3 Semantic distance computation

Based on the semantic characteristic functions, we develop a set of semantic measurement methods to measure semantic distances between two service nodes. We define that a semantic distance *dist* between two nodes S and R as

$$dist\,(\lambda(R), \lambda(S)) = \frac{\sum_{\forall \alpha \in \lambda} |\, \alpha(S) = \alpha(R)\,|}{|(\max(\lambda(R), \lambda(S)))|}$$

Here λ is the set of all the semantic characteristics functions.

The semantic measurement methods include a number of weighted formulas for computing the semantic distances between two service nodes, between two service pairs, and between two service sequences:

(1) $semantic\_distance(S, R) = \omega_s \cdot dist(\lambda(S), \lambda(R))$

(2) $semantic\_distance(S_1\text{-}S_2, R_1\text{-}R_2) = \omega_p \cdot dist(\lambda(S_1\text{-}S_2), \lambda(R_1\text{-}R_2))$

(3) $semantic\_distance(S_1\text{-}\ldots\text{-}S_n, R_1\text{-}\ldots\text{-}R_n) = \omega_f \cdot dist(\lambda(S_1\text{-}\ldots\text{-}S_n), \lambda(R_1\text{-}\ldots\text{-}R_n))$

Here ω_s, ω_p, ω_f are the weights for the individual, pair, and sequence semantic computations respectively.

A simple example is given below to illustrate how the algorithm works.

In order to construct a composite service to calculate the area of trapezium, one of the required services, an addition service, has to be located. The specification for this service is listed below:

- Number of inputs: **2**,
- Input data type: **double**,
- Output data type: **double**,
- Description keyword: **addition**,
- A mathematics calculation service.
- Can be used to calculate the perimeter of a rectangle

By applying the algorithm, we get the results which have been listed in table 1:

Matching Aspects	**Service1** double addition(double a, double b)	**Service2** int addition(int a, int b)	**Service3** double power(double a, double b)
IOPE	1.0	0.5	1.0
Ontology	1.0	1.0	1.0
Metadata	0.667	0.667	0.0
Upper Compositionality	1.0	1.0	0
Sum	3.667	3.167	2.0

Table 1: Semantic Matching Result of an Addition Service

The values in the table indicate the candidate services satisfaction rate for each aspect in the service requirement specification. From the above result, it is easy to see that the **Service1** has the highest overall satisfaction score which represents the shortest semantic distance to the requirement, so the Service1 will be the chosen service.

Related Work

Fensel et al. [9] proposed a web service modelling work (WSMF) to address how a service should be described. Roman et al. [12] proposed a web service modelling ontology (WSMO) based on the WSMF to semantically describe the aspects proposed in WSMF for describing services. Martin et al. [13] proposed a semantic mark-up language OWL-S to describe web services. This language describes a web service through three components: service profile, service process model, and service grounding. Medjahed et al. [14] provided a whole solution from semantic service description, service composability model, to automatic composition of web services. However, as we discussed previously, these research efforts did not consider enough contextual aspects for describing a service which is important in the process of service discovery and composition.

Conclusion and Future Work

Defining a contextual based semantic model to analyze and describe the structure and characteristics of services is very important in the research areas of semantic web service, grid computing, and web services. In this paper, we discussed how the web services can be composed from a conceptual perspective, proposed a contextual based semantic description model for automatic web service discovery and composition, and introduced a semantic matchmaking algorithm based the model to

accurately identify a service or a sequence of services. This is particularly significant in semantic based automatic service searches and matches as the number of services on the web is growing exponentially. Our next step is to develop an advanced quantitative analysis algorithm for semantic matching and to evaluate its performance. In order to achieve automatic service discover and composition, we also need to consummate the contextual based semantic model to more sufficiently describe services.

References

1. Foster, I. and Kesselman, C., The Grid: Blueprint for a New Computing Infrastructure, Publisher: Morgan Kaufmann Inc, 1999. ISBN: 1-555860-475-8
2. Foster, I., Kesselman, C., and Tuecke, S., The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications, 15 (3). 200-222. 2001
3. Foster, I., Kishimoto H., Savva, A., Berry, D., Djaoui, A., Grimshaw, A., Horn, B., Maciel, F., Siebenlist, F., Subramaniam, R., Treadwell, J., and Von Reich, J., The Open Grid Services Architecture, Version 1.0, Global Grid Forum, 2005 http://www.gridforum.org/documents/GFD.30.pdf
4. Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., 2001. Web Services Description Language (WSDL) 1.1 http://www.w3.org/TR/2001/NOTE-wsdl-20010315
5. W. Song, (2003) Semantic Issues in the Grid computing and Web Services (invited speech), in the Proceedings of International Conference on Management of e-Commerce and e-Government, Nanchang, China
6. Song, W., Du, X., and Munro, M., A Contextual based Conceptual Model and Similarity Computation Method for Automation of Web Service Discovery and Composition, Department report, University of Durham, UK, 2006
7. Dey, A. K. and Abowd, G. D., Towards a Better Understanding of Context and Context Awareness. Presented at the CHI 2000 Workshop on the What, Who, Where, When, Why and How of Context-Awareness, April 1-6, 2000
8. Martin, D., Paolucci, M., Mcllraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., and Sycara, K., 2004. Bringing Semantics to Web Services: The OWL-S Approach, presented at First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC), San Diego, California, USA
9. Fensel, D. and Bussler, C., The Web Service Modeling Framework WSMF. Electronic Commerce Research and Applications, Vol. 1, Issue 2, Elsevier Science B.V.
10. Lara, R., Lausen, H., Arroyo, S., Bruijn, J., and Fensel, D., Semantic Web Services: Description Requirements and Current Technologies, In International Workshop on Electronic Commerce, Agents, and Semantic Web Services, September 2003.
11. DAML-S: Semantic Markup for Web Services (The DAML Services Coalition), 2001 http://www.daml.org/services/daml-s/2001/10/daml-s.html
12. Roman, D., Keller, U., Lausen, H., Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., and Fensel, D., Web Service Modelling Ontology (WSMO), WSMO Final Draft April 13, 2005. http://www.w3.org/Submission/WSMO/
13. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., Mcllraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., and Sycara, K., 2004. OWL-S: Semantic Mark-up for Web Services. http://www.daml.org/services/owl-s/1.1/overview/
14. Medjahed, B., Bouguettaya, A., and Elmagarmid, A. K., Composing Web Services on the Semantic Web, the VLDB Journal 2003. Volume 12, pp. 333-351
15. Berners-Lee, T., Hendler, J., and Lassila, O., The Semantic Web, Scientific American, May 17, 2001, pp. 35-43
16. De Roure, D., Jennings, N. R., and Shadbolt, N. R. The Semantic Grid: Past, Present and Future. Procedings of the IEEE, 2005, Volume 93, Issue 3, pp. 669-681
17. W3C Web Service Definition http://www.w3.org/TR/ws-arch/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Annotating scientific data: why it is important and why it is difficult.

Rajendra Bose
University of Edinburgh

Peter Buneman
University of Edinburgh

Denise Ecklund
Objective Technology Group

Abstract

Annotation of existing data is becoming a standard tool in many branches of e-science. Increasingly, databases are being built to receive annotation, and other tools are being developed to annotate existing databases. Annotation is becoming an important part of communication among scientists. In this paper we review various kinds of annotation systems and describe the importance of designing databases in such a way that they can receive annotation. This includes designing *extensible* databases and the need for some form of *co-ordinate system* for the attachment of annotations.

1 Annotation: adding to existing structure

Most people will agree with the dictionary definition of *annotation* as the process of adding comments or making notes on or upon something. Such notes have traditionally served a variety of purposes, including explaining, interpreting or describing some underlying text. Annotation is often for personal use but, more importantly in our context, it can be a means of disseminating useful information. For example, annotated bibliographies and textual criticism are well understood uses of annotation for dissemination of knowledge. Annotation of images and plans is also commonplace; much of cartography is about spatial annotation.

The use of on-line, digital data has caused a revolution in the way scientific research is conducted. In every area of science, much investigation now depends not on new experiments, but on databases in which experimental evidence has been stored. However, this evidence is seldom raw experimental data; it is typically some form of interpretation of the data, and annotation is an increasingly important part of that interpretation. Nowhere is this more apparent than in molecular biology, where the value of some databases lies almost entirely in the annotation they add to data extracted from other databases. This added value often represents substantial investment of effort. One example is UniProt (Universal Protein Knowledgebase) [ABW+04], which is supported by upwards of 100 of curators or annotators. There is also an increasing amount of machine-generated annotation: pattern recognition and machine learning techniques are being used in biology and astronomy to flag suspect data.

In contrast to annotation within databases, other forms of annotation are externally affixed "over" a body or collection of data similar to the way sticky notes are now attached to PDF documents and web pages. [MD99], discusses "superimposed information" – 'data placed over existing [base] information sources to help organise, access, connect and reuse information elements in those sources."

The importance of annotation was, as with many so many other issues, recognised as important by Vannevar Bush [Bus45] who says "A record, if it is to be useful to science, must be continuously extended, it must be stored, and above all it must be consulted." Annotation is ubiquitous on the Web – in Wikis, review/opinion sites, newsgroups, etc. It is now a basic activity in the publication of scientific and scholarly data. It is therefore essential that the database community and the whole community of

digital publishers obtain some understanding of this process and the associated pitfalls and technological requirements.

1.1 A framework for annotation

In order to compare various types of annotation systems we suggest an informal framework that consists of the following basic components:

An annotation is some set of data elements that is added to an existing base or target that possesses structure. In order to create an annotation, some form of attachment point is used implicitly or explicitly. We shall use the term *co-ordinate system* for the mechanism for describing the attachment point. Some care is needed, during database design, in making sure that the co-ordinate system is durable. Moreover one frequently finds several co-ordinate systems in simultaneous use. Understanding the mappings between the co-ordinate systems is seldom straightforward. Let us consider some examples (which are discussed in more detail later in this paper):

- Cartographic data. The use of multiple co-ordinate systems (such as longitude and latitude vs. a local grid) is commonplace in cartography and mappings between such systems are well understood. The point of attachment to a image representation of a map is specified by such a co-ordinate system. However, recent map data now relies on some form of object-oriented representation of cartographic features, and attachment is, presumably, to some object identifier. Note that there is a subtlety about what is being annotated; moreover the correspondence between the two co-ordinate systems is no longer a simple 1-1 mapping.
- Molecular biology. This has moved in the reverse direction. The original co-ordinate systems were the gene identifiers used in the various databases. Only recently have the linear (chromosome, offset) co-ordinates determined by genetic sequencing been discovered and, once again, the mapping is not 1-1.

The various aspects of annotation are illustrated in Figure 1. The genome column summarises two of the co-ordinate systems in use in that domain. The HBP column shows that while the co-ordinate system may be simple, the attachment process is not. AstroDAS is interesting in this context because its purpose is precisely to reconcile the co-ordinate systems in a variety of database. It is a database in which the annotations of the objects are the co-ordinates (typically relational keys) of objects in other catalogues. The intention of this annotation is to support the more general forms of cross-database annotation, which we describe below.

In the remainder of the report, we present a series of examples of scientific annotation in Section 2, and refer to our basic framework to help compare them. In Section 3 we discuss some key concepts of database annotation and suggest them as topics for further research.

2 Examples of scientific annotation systems

2.1 UniProt database annotation

Perhaps the most well-known examples of database annotation are to be found in bioinformatics and in the design of information systems like UniProt, which consists of an assemblage of databases including Swiss-Prot, an annotation database produced by specialist curators, and TrEMBL, which provides automated annotations for proteins (`http://www.ebi.uniprot.org/about/background.shtml`). These databases were created to to disseminate protein sequence data and associated analyses. They were designed specifically to receive annotation.

The curators of Swiss-Prot [BA00] are quite specific about which fields in the database they regard as annotation and which are "core data". In figure 2, the boxed areas are those classified as annotation, the publication and taxonomic entries, perhaps because the Swiss-Prot organisation was not responsible for its creation, are regarded as core data. This database illustrates a number of interesting aspects of annotation which we discuss further in Section 3.1. Note that several of the fields have "pointers" to entries in other databases, which provide mappings between co-ordinate systems.

	HBP image annotation	**Astrodas**	**Genome(human)**	
Annotation target:				
what	brain image	celestial object in astronomy catalogue	gene	
structure	pixel array	catalogue (RDBMS schema)	database schema	sequence
co-ordinate system	(x, y) co-ordinates of pixel	catalogue + object id (relational key)	object id ("accession number")	chromosome + offset
Annotation:				
what	domain-specific term (ontology element)	mappings to other catalogues	free and structured text	
location of attachment	2D contour	catalogue+id	key + attribute	chromosome + start/stop positions
purpose	link Web-accessible images to and find instances	share assertions across different catalogues	general comments and classification	

Figure 1. Comparison of annotation systems

2.2 Genome annotation

In contrast to Swiss-Prot, BioDAS [SED02] is an external annotation system for a variety of databases. The Distributed sequence Annotation System (DAS, later BioDAS) protocol [SED02] was designed to serve this purpose. The architecture is that of an "open" client-server annotation system communicating via an extension of HTTP; significantly, the addition of new annotation servers requires only minimal coordination between data providers.

BioDAS includes a client capable of requesting both (1) the coordinate system or "reference map" of base pairs for a specific genome from a reference server, and (2) a set of uniquely identified sequence annotations, anchored to the reference map by start and stop values, from an annotation server [DJD+01]. The client requests are URLs that are constructed according to simple conventions in an HTTP request; the servers respond to these requests with a Generic (genomic) Feature Format (GFF)-derived XML document [Ens06]. The Ensembl Genome Browser web application (`www.ensembl.org`) employs DAS functionality.

IBM developerWorks uses a similar client/server architecture to provide a general solution for annotation of digital data (`http://www-106.ibm.com/developer\-works/webservices/library/ws-annotation.html`). In this scenario, an annotation is an XML document that is linked to a target data object (for example, a database, word processing document or spreadsheet) with a unique preexisting identifier (or one generated by a hash value). They define an Annotation Web services API consisting of methods for communication between an annotation client and server for creating, updating, and retrieving annotations and annotation structure definitions. [Wei03] refers to a system to store and retrieve annotation for the drug discovery process based on the IBM InsightLink product which contains an implementation of the Annotation Web services API.

Other systems exist for annotating genomic data. The SEED project [RDS04] is similar to BioDAS, but more ambitious in infrastructure. The project focuses on allowing an individual researcher to perform rapid gene sequence annotation, to integrate his private data with public databases during the annotation process, and to view annotation for related biological function across many organisms rather than for just one organism.

MyGrid [ZGG+03] includes projects that use a graphical workflow editor to assist bioinformatics researchers in using a series of annotation-related web services during the process of annotating a genome sequence. This work also experiments with semi-automatic semantic labelling of annotation workflows.

```
ID   11SB_CUCMA     STANDARD;      PRT;   480 AA.
AC   P13744;
DT   01-JAN-1990, integrated into UniProtKB/Swiss-Prot.
DT   01-JAN-1990, sequence version 1.
DT   21-MAR-2006, entry version 51.
DE   11S globulin beta subunit precursor [Contains: 11S globulin gamma
DE   chain (11S globulin acidic chain); 11S globulin delta chain (11S
DE   globulin basic chain)].
OS   Cucurbita maxima (Pumpkin) (Winter squash).
OC   Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta;
OC   Spermatophyta; Magnoliophyta; eudicotyledons; core eudicotyledons;
OC   rosids; eurosids I; Cucurbitales; Cucurbitaceae; Cucurbita.
OX   NCBI_TaxID=3661;
RN   [1]
RP   NUCLEOTIDE SEQUENCE [MRNA].
RC   STRAIN=cv. Kurokawa Amakuri Nankin;
RX   MEDLINE=88166744; PubMed=2450746;
RA   Hayashi M., Mori H., Nishimura M., Akazawa T., Hara-Nishimura I.;
RT   "Nucleotide sequence of cloned cDNA coding for pumpkin 11-S globulin
RT   beta subunit.";
RL   Eur. J. Biochem. 172:627-632(1988).
RN   [2]
RP   PROTEIN SEQUENCE OF 22-30 AND 297-302.
RA   Ohmiya M., Hara I., Mastubara H.;
RT   "Pumpkin (Cucurbita sp.) seed globulin IV. Terminal sequences of the
RT   acidic and basic peptide chains and identification of a pyroglutamyl
RT   peptide chain.";
RL   Plant Cell Physiol. 21:157-167(1980).
CC   -!- FUNCTION: This is a seed storage protein.
CC   -!- SUBUNIT: Hexamer; each subunit is composed of an acidic and a
CC       basic chain derived from a single precursor and linked by a
CC       disulfide bond.
CC   -!- SIMILARITY: Belongs to the 11S seed storage protein (globulins)
CC       family.
CC   ---------------------------------------------------------------------
CC   Copyrighted by the UniProt Consortium, see http://www.uniprot.org/terms
CC   Distributed under the Creative Commons Attribution-NoDerivs License
CC   ---------------------------------------------------------------------
DR   EMBL; M36407; AAA33110.1; -; mRNA.
DR   HSSP; P04776; 1FXZ.
DR   InterPro; IPR006045; Cupin_1.
DR   InterPro; IPR007113; Cupin_region.
DR   InterPro; IPR011051; Cupin_RmlC_type.
DR   InterPro; IPR006044; Seedstore11s_pln.
DR   Pfam; PF00190; Cupin_1; 2.
DR   PRINTS; PR00439; 11SGLOBULIN.
DR   PROSITE; PS00305; 11S_SEED_STORAGE; 1.
KW   Direct protein sequencing; Pyrrolidone carboxylic acid;
KW   Seed storage protein; Signal; Storage protein.
FT   SIGNAL        1     21
FT   CHAIN        22    296       11S globulin gamma chain.
FT                                 /FTId=PRO_0000032028.
FT   CHAIN        22    480       11S globulin beta subunit.
FT                                 /FTId=PRO_0000032027.
FT   CHAIN       297    480       11S globulin delta chain.
FT                                 /FTId=PRO_0000032029.
FT   MOD_RES      22     22       Pyrrolidone carboxylic acid.
FT   DISULFID    124    303       Interchain (between gamma and delta
FT                                chains) (Potential).
FT   CONFLICT     27     27       S -> E (in Ref. 2).
FT   CONFLICT     30     30       E -> S (in Ref. 2).
SQ   SEQUENCE   480 AA;  54626 MW;  BCD8A83DD1AED93C CRC64;
     MARSSLFTFL CLAVFINGCL SQIEQQSPWE FQQSEVWQQH RYQSPRACRL ENLRAQDPVR
     RAEAEAIFTE VWDQDNDEFQ CAGVNMIRHT IRPKGLLLPG FSNAPKLIFV AQGFGIRGIA
     IPGCAETYQT DLRRSQSAGS AFKDQHQKIR PFREGDLLVV PAGVSHWMYN RQQSDLVLIV
     FADTRNVANQ IDPYLRKFYL AGRPEQVERG VEEWERSSRK GSSGEKSGNI FSGFADEFLE
     EAFQIDGGLV RKLKGEDDER DRIVQVDEDF EVLLPEKDEE ERSRGRYIES ESESENGLEE
     TICTLRLKQN IGRSVRADVF NPRGGRISTA NYHTLPILRQ VRLSAERGVL YSNAMVAPHY
     TVNSHSVMYA TRGNARVQVV DNFGQSVFDG EVREGQVLMI PQNFVVIKRA SDRGFEWIAF
     KTNDNAITNL LAGRVSQMRM LPLGVLSNMY RISREEAQRL KYGQQEMRVL SPGRSQGRRE
//
```

Figure 2. An entry from UniProt

2.3 Annotating biomedical images

Some systems are designed to create web-accessible collections of annotated biomedical images. Gertz et al. [GSG+02] develop a graph model of annotations for use in the Human Brain Project (HBP): annotation nodes serve to connect specific image region of interest nodes with concept nodes from a controlled vocabulary. Graph edges define the relationship between nodes; one such relationship is "annotation of". They also develop a framework for querying annotation graphs based on path expressions and predicates, which they test in a prototype system. Column (1) of Figure 1 refers to HBP image annotation.

The Edinburgh Mouse Atlas Project (EMAP) involves two types of annotations for images. EMAP provides annotations that make connections between both a standard anatomical nomenclature and the results of tissue-level gene expression experiments with regions of 3D mouse embryo tissue images and 2D tissue slices. This project provides a suite of tools, including an interactive website (`http://genex.hgu.mrc.ac.uk/intro.html`). The tools allow one to browse text nomenclature and make queries about gene expressions that return sets of images or a list of genes expressed for a given embryo image. Another way to query for gene expressions is to interactively select an area of a 2D image. EMAP involves centralised editorial control and curation; an editorial review board decides whether to accept gene expression experiment results, and regions of images are manually coloured by an expert.

2.4 AstroDAS: Annotating astronomy catalogues

Over the past several decades, databases or catalogues of celestial object observations, recorded by disparate telescopes and other instruments over various time periods, have migrated online. Central to the astronomical community's concept of a global "Virtual Observatory" is the ability to identify records in these different catalogues as referring to the same celestial object. Because the recorded location of a celestial object may vary slightly from catalogue to catalogue due to unavoidable measurement error at the instrument level, the general catalogue matching problem cannot be solved by spatial proximity alone, and some researchers develop their own complex algorithms for matching celestial objects across different catalogues.

To provide astronomers with the ability to share their assertions about matching celestial objects directly with their colleagues, we have created prototypes for AstroDAS, a distributed annotation system partly inspired by BioDAS [BMPR06]. AstroDAS features an annotation database with a web service interface to store and query annotation, and resolves queries on astronomy catalogues using mapping tables that are dynamically constructed from annotations of celestial object matches. The AstroDAS prototypes complement the existing OpenSkyQuery system for distributed catalogue queries.

The ultimate aim of AstroDAS is similar to the goal of the earlier BioDAS: to record and share scientific assertions with a wider com-

munity. Whereas biologists use annotation in BioDAS to interpret the DNA sequences in a genome, however, astronomers seek to share the mapping of entities derived from their research across established scientific databases. Specifically, astronomers want to be able to share their identification of matching celestial objects within the existing federation of disparate catalogues.

3 Concepts and research topics in database annotation

One of the most useful effects a report such as this could have would be to help the designers of a new database, schema or data format to prepare their data for annotation. Of course, some databases, especially those in bioinformatics, are designed to receive annotation. But we have seen many examples of the need to accommodate *ad hoc* annotation and the need for ad hoc annotation to migrate to a more systematic form of annotation, that is, to become part of the regular database structure, which we discuss further in the following sections. We also discuss annotation queries, research topics in relational database annotation, and annotating annotations.

3.1 Annotation and the evolution of database structure

We return to the Swiss-Prot example of annotation within databases: Figure 2 shows a single entry in Swiss-Prot. It is debatable what one should classify as data, metadata or annotation. However, from a database perspective, the entry illustrates several interesting points, including the evolution of structure. The structure of the entry is an old, purpose-built file format with a two-letter code giving the meaning of each line of text. Notice that the comment lines (`CC`) have become structured with entries of the form `-!- FUNCTION: ...` which provide a degree of machine-readability of the comment text. These entries were presumably not anticipated by the designers of the original format, and the alternative of specifying some further two-letter codes for these entries, was presumably ruled out as it would confuse existing software designed to parse the format. There are now 26 such subfields, one of which has additional machine-readable internal structure. The important observation here is that annotation plays an important part in the evolution of both the form and content of data. What was once unknown or regarded as *ad hoc* annotation has become part of the database structure. It is almost certainly the case that the curators of Swiss-Prot now make extensive use of database technology and that what is exported in Figure 2 is a "rendering" or database view of the internal data. While database management systems provide some help with structural evolution, it is always problematic. In this respect, databases designed with conventional (relational or object-based) structuring tools offer better prospects for extensibility than XML structured with DTDs or XML-Schema which are, at heart, designed to express the serialisation of data.

3.2 Location and attachment of annotations

The annotations in the `CC` fields in Figure 2 appear to refer to the entire Swiss-Prot entry. Reading down, one finds feature table (`FT`) lines that contain "fine-grain" annotation about different segments of the sequence data. There is a subtle difference between the two forms of annotation. The `CC` annotations are understood to refer to the whole entry because they occur *inside* that entry. The `FT` annotations are *outside* the structure being annotated and therefore require extra information, in this case a pair of numbers specifying a segment, to describe their attachment to the data. Notice that this assumes a *stable* co-ordinate system. If the sequence data were to be updated with deletions or insertions, attachment of annotations would be problematic.

Consider another, fanciful, example of a fine-grain attachment in which one wants to say something like "The third author of the first citation also publishes under the alias John Doe". One could imagine inserting this text in the text of the Reference Author (`RA`) line, but this is likely to interfere with any software that parses this line. Alternatively one could place it externally in some other field of the entry. Once again, this assumes that the co-ordinate system is stable. For example, it assumes that the numbering of the citations does not change when the database is updated.

Another issue is the attachment of an annotation to several entries/objects in any of the

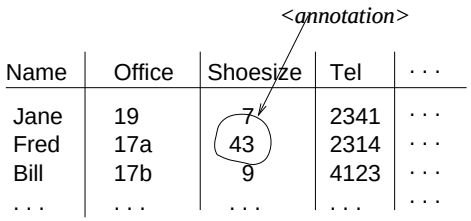


Figure 3. A simple annotation

databases we are considering. One could place the same annotation (with references to all relevant entries) in each of the relevant entries, but this is a standard example of "non-normalised" data. The solution is to build a separate annotation table, or "stand off markup" [TM97] with links to the appropriate entries. Again, this requires an extension to the existing structure of the database.

We have already noted that annotations are sometimes placed inside the annotated object and sometimes outside and that many annotations are, for reasons of database security, necessarily stored externally. External annotations require a co-ordinate system in order to specify how they are to be attached to the data. It is worth a brief digression not observe that the point of attachment does not tell us everything. Consider the annotation of one value of the table shown in Figure 3. and consider some possibilities for ⟨*annotation*⟩:

1. This is a prime number
2. This is probably a European shoe size
3. This is way too big (for a shoe size)
4. This is way too big (for Fred)
5. The normal range is 5-14

All of these are perfectly valid annotations, but the referent requires some explanation. In (1) the annotation has nothing to do with the location; it is an annotation on the value that could be attached to any occurrence of the number 43. By contrast, in (2) the annotation has to do with the column (or domain) and could reasonably be attached to any other occurrence of 43 in the Shoesize column. Similarly for (3), though this is less informative. The only annotation that is specifically about the relationship between the value, 34, and the location, the Shoesize field of the Fred tuple, is (4). Finally, (5) is an annotation that should be attached to the schema, rather than the data; however the schema is frequently transformed in views of the data, and the attachment of such annotations may be problematic.

To return to the specification of attachment of external annotations, consider first how one would specify the attachment in Figure 3. One would provide the name of the table, identifier for the tuple, and the name (Shoesize) of the field within the tuple. The tuple identifier could be a key, or it could be the internal tuple identifier provided by the database management system. It is regarded as bad practice to modify a key and it is impossible to change an internal tuple identifier (they last for the lifetime of a tuple and are never reused). Thus the (table name, tuple identifier, field name) triple should serve as a stable "co-ordinate system" for attachment in a well-defined relational database.

The same idea can be extended to hierarchically structured data such as XML; the details are straightforward [BDF+02] and are not given here. The point is first that the designers of new data sets should not only describe the schema, they should also describe a co-ordinate system for the attachment of annotations. Second, if the data set is updated, the updates should respect the co-ordinate system. One should not, for example, recycle identifiers or field names.

3.3 Querying annotations

Work in the ediKT project at Edinburgh (`http://www.edikt.org`) with the Edinburgh Mouse Atlas Project (EMAP) suggests that users of the mouse atlas want to be able to query annotations for two distinct purposes: (1) to locate annotations where the annotation values themselves are of interest ("show me all annotations which have a value of 'gene expression pattern X'"); and (2) to locate annotations where the associated base data values are of interest ("show me all the annotations associated with the following mouse atlas images"). Many existing annotation systems provide only a limited ability to query over annotation values. For example, consider systems for web page annotation: queries on this type of annotation might be limited to *find* capabilities supported in the client browser.

Supporting annotation queries for case (1) is more likely to be straightforward than case (2). For the second case one needs to know *where* the annotation is attached to the base data and perhaps *why* it is attached. How this is captured in the database and expressed in the query is an open question.

3.4 Annotating relational databases: recent work

Relational databases have had an extraordinarily successful history of commercial success and fertile research. It is not surprising, therefore, that database researchers would first attempt to understand annotation in the context of relational databases. One of the immediate challenges here is to understand how annotations should propagate through queries. If one thinks of annotation as some form of secondary mark-up on a table, how is that mark-up transferred to the result of a query. If, for example, an annotation calls into question the veracity of some value stored in the database, one would like this information to be available to anyone who sees the database through a query of *view*.

Equally important is the issue of backwards propagation of annotations. We consider, as a loose analogy, the BioDAS system, based on the DAS system discussed in Section 2.2. The users see and annotate the data using some GUI, which we can loosely identify with a database view. The annotation is transferred backwards from the GUI to an annotation on some underlying data source and is then propagated forwards to other users of the same data. Following the correspondence, the question is how does an annotation propagate through a query both backwards and forwards?

It is easy to write down the obvious set of rules for the propagation of annotation through the operations of the relational algebra. However, because of nature of relational algebra, inverting these rules is non-deterministic. An annotation seen in the output could have come from more than one place in the input. To take one example: suppose one places an annotation on some value in the output of a query Q. Of all the possible annotations on the source data (the tables on which Q operates) is there one which causes the desired annotation – and only that annotation – to appear in the output of Q. The complexity of this and several related annotation problems have been studied in [BKT02] which also shows the connection with the view deletion problem.

In [BCTV04] a practical approach is taken to annotation in which an extension of SQL is developed which allows for explicit control over the propagation of annotations. Consider the following simple join query

$$\begin{array}{ll} \mathsf{SELECT} & R.A,\ R.B,\ S.C \\ \mathsf{FROM} & R, S \\ \mathsf{WHERE} & R.B = S.B \end{array}$$

Suppose the source is annotated. Presumably an annotation on a B value of R should propagate to the B field of the output, because $R.B$ is given as the output. But should an annotation on a B field of S also be propagated to the B field of the output? The structure of the SQL indicates that it should not, but the query obtained by replacing the first line by

$$\mathsf{SELECT}\ R.A,\ S.B,\ S.C$$

is equivalent, so maybe the answer should be yes. The idea in [BCTV04] is to allow the user to control the flow of annotation by adding some further propagation instructions to the SQL query. The paper shows how to compute the transfer of annotations for the extended version of SQL and demonstrates that for a range of realistic queries the computation can be carried out with reasonable overhead.

The work we have described so far has been limited to annotating individual values in a table. Recently Geerts *et al.* [GKM06] have taken a more sophisticated approach to annotating relational data. What they point out is is that it is common to want to annotate *associations* between values in a tuple. For example, in the query above one might want to annotate the A and B fields in the output with information that they came from input table R and the B and C fields with information that they came from table S. To this end the introduce the concept of a *block* – a set of fields in a tuple to which one attaches an annotation and a *colour* which is essentially the content or some property of the annotation. They also investigate both the theoretical aspects and the overhead needed to implement the system. However, as we have indicated in Section 3.2 that attachment may be even more complex, requiring associations between data and schema, for example.

3.5 Provenance and Annotation

The topic of *data provenance* is of growing interest and deserves separate treatment. However there are close connections with annotation. One view of the connection is that provenance – information about the origins of a piece of data – is simply another form of annotation that should be placed on data. It is certainly true that there are many cases where provenance information is added after the creation of data. However, it would be much better if provenance were captured automatically, in such a way that it becomes an intrinsic part of the data.

A more interesting connection is to be found in [BCTV04] and related papers. Much data in scientific databases (e.g. the "core data" in Figure 2) has been extracted from other databases. If the data in the source database has been annotated, surely the annotations should be carried into the receiving database. If the receiving database is a simple view of the source data, then the mechanisms described in [BCTV04], or some generalisation of them, should describe both provenance and how annotations are to be copied. However, manually curated databases are more complex than views, and in this case understanding the movement of annotations is still an open problem.

4 Conclusions

Although we have not done enough work to substantiate this claim, we believe it likely that most of the 858 molecular biology databases listed in [Gal06] involve some form of annotation. Moreover, as we have tried to indicated, annotation is of growing importance in other areas of scientific research. The success of new databases will depend greatly on the degree to which they will support annotation. In this respect, the following points are crucial both in database design and in systems architecture:

- the provision of a co-ordinate system to support the attachment of annotations,
- the linkage or mapping of that co-ordinate system to other, existing, co-ordinate systems, and
- the need for extensibility in databases that are designed to receive annotations.

In each of these areas, there is further research needed. Moreover, annotations often express complex relationships between schema and data. To bring this into a uniform framework is a challenge for both database and ontology research.

5 Acknowledgements

We would like to thank Mark Steedman, Robert Mann, Amos Storkey, Bonnie Webber, as well as the members of the Database Group in the School of Informatics. This survey has been supported in part by the Digital Curation Centre, which is funded by the EPSRC e-Science Core Programme and by the JISC.

References

[ABW+04] R. Apweiler, A. Bairoch, C.H. Wu, W.C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M.J. Martin, D.A. Natale, C. O'Donovan, N. Redaschi, and L.S. Yeh. Uniprot: the universal protein knowledgebase. *Nucleic Acids Res*, 32:D115–D119, 2004.

[BA00] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL. *Nucleic Acids Research*, 28:45–48, 2000.

[BCTV04] Deepavali Bhagwat, Laura Chiticariu, Wang Chiew Tan, and Gaurav Vijayvargiya. An annotation management system for relational databases. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, pages 912–923, Toronto, Canada, 2004. Morgan Kaufmann.

[BDF+02] Peter Buneman, Susan Davidson, Wenfei Fan, Carmem Hara, and Wang-Chiew Tan. Keys for XML. *Computer Networks*, 39(5):473–487, August 2002.

[BKT02] Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. On the Propagation of Deletions and Annotations through Views. In *Proceedings of 21st ACM Symposium on Principles of Database Systems*, Madison, Wisconsin, 2002.

[BMPR06] R. Bose, R. Mann, and D. Prina-Ricotti. Astrodas: Sharing assertions across astronomy catalogues through distributed annotation. In *International Provenance and Annotation Workshop (IPAW 2006)*, pages 193–202, Chicago IL, 2006. Springer, LNCS 4145.

[Bus45] V. Bush. As we may think. *The Atlantic Monthly*, June 1945.

[DJD+01] R Dowell, R Jokerst, A Day, S Eddy, and L Stein. The distributed annotation system. *BMC Bioinformatics*, 2(7), 2001.

[Ens06] Ensembl: Information: Data: External data: About the distributed annotation system (das), 2006. http://www.ensembl.org/info/data/external_data/das/index.html.

[Gal06] Michael Y. Galperin. The molecular biology database collection: 2006 update. *Nucleic Acids Research*, 34:D3–D5, 2006,. Database issue.

[GKM06] Floris Geerts, Anastasios Kementsietsidis, and Diego Milano. Mondrian: Annotating and querying databases through colors and blocks. In *ICDE*, page 82, 2006.

[GSG+02] Michael Gertz, Kai-Uwe Sattler, Frederic Gorin, Michael Hogarth, and Jim Stone. Annotating scientific images: A concept-based approach. In Jessie Kennedy, editor, *14th International Conference on Scientific and Statistical Database Management (SSDBM 2002)*, pages 59–68, Edinburgh, Scotland, 2002. IEEE Computer Society.

[MD99] D. Maier and L. Delcambre. Superimposed information for the internet. In *WebDB 1999 (Informal Proceedings)*, pages 1–9, 1999. http://www-rocq.inria.fr/ cluet/WEBDB/maier.pdf.

[RDS04] R.Overbeek, T. Disz, and R. Stevens. The SEED: a peer-to-peer environment for genome annotation. *Comm. ACM*, 47(11):46–51, 2004.

[SED02] Lincoln D. Stein, Sean Eddy, and Robin Dowell. Distributed sequence annotation system (das) specification version 1.53. Technical report, 21 March 2002 2002.

[TM97] Henry S. Thompson and David McKelvie. Hyperlink semantics for standoff markup of read-only documents. In *SGML '97 Conference Proceedings*, pages 227–229, Barcelona, Spain, 1997.

[Wei03] H.J.R. Weintraub. The need for scientific data annotation. *Abstracts of Papers of the American Chemical Society*, 226:303–304, 2003.

[ZGG+03] J. Zhao, C. Goble, M. Greenwood, C. Wroe, and R. Stevens. Annotating, linking and browsing provenance logs for e-science. In *Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*, Sanibel Island, Florida, 2003. Online proceedings (at ISWC 2003).

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

The OMII Software Distribution

Justin Bradley, Christopher Brown, Bryan Carpenter, Victor Chang, Jodi Crisp, Stephen Crouch, David de Roure, **Steven Newhouse**, Gary Li, Juri Papay, Claire Walker, Aaron Wookey

Open Middleware Infrastructure Institute (OMII)
University of Southampton

Abstract

This paper describes the work carried out at the Open Middleware Infrastructure Institute (OMII) and the key elements of the OMII software distribution that have been developed within the community through our support of the open source development process by commissioning software. The main objective of the OMII is to preserve and consolidate the achievements of the UK e-Science Programme by collecting, maintaining and improving the software modules that form the key components of a generic Grid middleware. Recently, the activity at Southampton has been extended beyond 2009 through a new project, OMII-UK, which forms a partnership that now includes the OGSA-DAI activities at Edinburgh and the [my]Grid project at Manchester.

1 Introduction

In this paper we summarise the results achieved by the Open Middleware Infrastructure Institute (OMII). Over the last two years the OMII model has become firmly established itself in the Grid domain with similar projects aimed at the consolidation of Grid middleware investment emerging in Europe (OMII-Europe [OMII-Europe]) and China (OMII-China [OMII-China]) and the established NMI (NSF Middleware Initiative) in the United States. The objectives of the OMII project can be summarised by the following points:

- Creating a one-stop portal and software repository for open-source Grid middleware, including comprehensive information about its function, reliability and usability.
- Provide quality-assured software engineering, testing, packaging and maintenance of software in the OMII repository, ensuring it is reliable, portable and easy to both install and use.
- Lead the evolution of Grid middleware at international level, by commissioning open-source software development and wide-reaching collaboration with industry and relevant standards bodies.
- Distribute a sustained, well-engineered, interoperable, documented and supported set of easily-used integrated middleware services, components and tools.
- Engage proactively with user communities by listening and responding carefully to their requirements and comments.

In this paper we describe in Section 2 the software engineering process adopted at the OMII. Section 3 details the OMII software stack of Grid components. Section 4 summarises the paper.

2 The Software Engineering Process

This section describes the software engineering process that has been implemented within OMII (see Figure1). This process is in daily use at the OMII and forms the backbone of the operation.

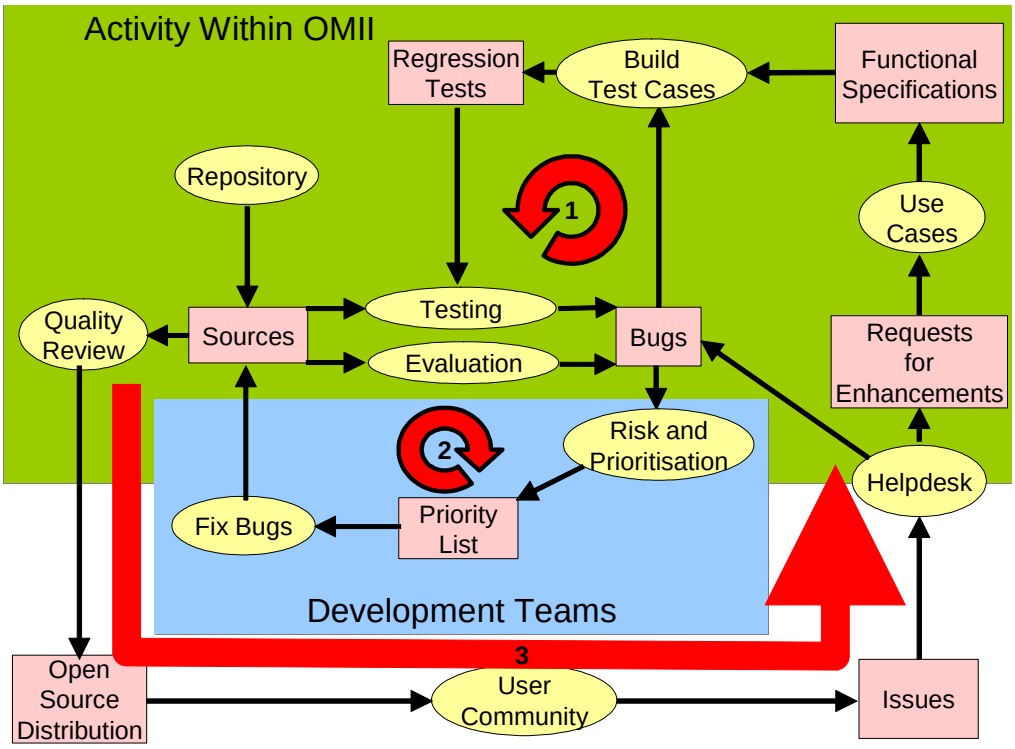


Figure 1 - Software engineering process

The software engineering process comprises three interlinked loops:

Loop 1 is represented by the Testing loop. The Testing loop evaluates the incoming software against test cases generated from their functional specifications and discovers bugs, defined as deviations from the functional specification, in the implementation or the test suite. This is the core 'quality improvement' cycle of the software engineering process involving detailed checking of the source code, with a daily-build regression testing process that uses a constantly evolving set of regression tests. These tests are composed of existing unit tests and integration tests of the deployed system. This activity generates new bugs, whose existence is recorded in the bug tracking system for later review.

A key pre-requisite to building regression tests is to have a clear functional specification describing the code, the standards to be complied with and the environment that the software has to function in (which changes constantly), and associated documentation describing the operation of the software. The functional specification provides a starting point for code review and the generation of test cases (see Figure 2).

As the code and functional specification (and user documentation) are verified in isolation, the OMII gains twice the benefits. The code is independently checked to prove it does what it is supposed to; likewise test cases derived from the functional specification independently check that same functionality. These two tasks are carried out by separate teams who deliberately do not collaborate until later on in the process.

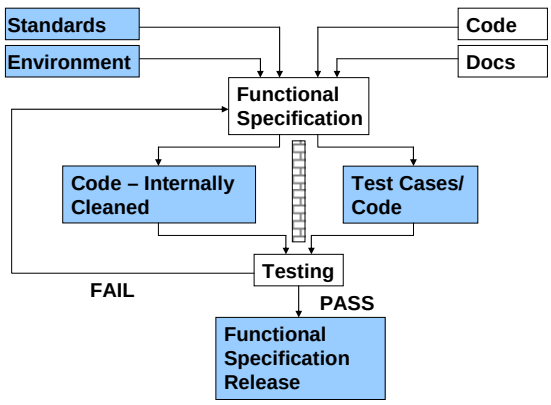


Figure 2 – The software enhancement process

For sources submitted to the OMII, some test cases may already exist. If not, or if test coverage is deemed inadequate, new test cases will be required from the external development team or developed internally. The types and quality of test cases are also reviewed and new test cases developed as a result of bug discovery. The high-level test cases and testing tasks are added at an early stage to the 'High-Level Test Plan' to provide feedback into the design process.

Testing forms a very large part of the OMII quality function and is carried out on *all* code with the aim of continuously improving the quality of the product. Several types of testing already happen at the OMII and the range and scope of tests (e.g. usability, standards compliance, etc.) is always being expanded.

Loop 2 characterises the development process. Within this process the identified issues are assessed for risk and prioritised for fixing in the next release cycle by the many development teams working within the OMII organisation. Bugs are discovered as a result of both internal testing, internal evaluation, from the development teams and as a result of the external use of the OMII distribution. In collaboration with the appropriate development team, OMII will assess the risk and impact of fixing a bug. Prioritisation takes place at the start of the development cycle and on a continuous basis within the OMII's bug-tracking system. As the resources required to fix all bugs, will inevitably exceed those available, maximum benefit must be obtained from the available resources in order to ship code of the very highest quality within any given timeframe. When the fix has been made, it will be reviewed. Normally this is done using a peer-review approach. Once the fix is agreed, the code will be checked into the code repository under the specified bug id prior to re-testing. Cross-fertilisation of developer skills is ensured by encouraging developers to fix code in areas of the code base that they are less familiar with. This prevents a single developer being the only expert in a particular section of code. The OMII development team at Southampton performs weekly 'bug-scrubs' which prioritise the bugs for fixing, taking into account the risks associated with the fix. The activity of fixing bugs generates new versions of the source modules, and is another opportunity to check the source directly rather than just by testing.

Loop 3 describes the public use of the software. The contributed sources, having reached a satisfactory quality, are then integrated into the public distribution and released to the community. Issues are identified as bugs that require fixing, or feature requests requiring enhancements to the functional specification. Problems raised by external users initially get entered in the OMII Helpdesk where, depending on their nature, they may go on to become a bug in the OMII bug-tracking system or considered as proposals for enhancements for the next release cycle.

3 OMII Software Architecture

The key components of the OMII software architecture are presented in Figure 3. Some of these components are already part of the OMII distribution, other we expect to integrate over the next 6-12 months.

The OMII software distribution is based around a lightweight client and server distribution. Both contain standalone tools and a web service based infrastructure currently based upon the Apache Axis toolkit. We expect to provide support for the deployment of web portals, through the provision of a JSR168 portlet compliant hosting environment. We will continue to source application and infrastructure services from the community by commissioning open-source software development.

Much of the commissioned open-source software development activity supports on-going or emerging standards activity – either in W3C (World Wide Web Consortium), OASIS (Organisation for the Advancement of Structured Information) or the Open Grid Forum (previously the Global Grid Forum). We see this as a vital part of our community support. We expect to source and integrate additional software components through partnerships with other Grid middleware providers, such as gLite and GT4.

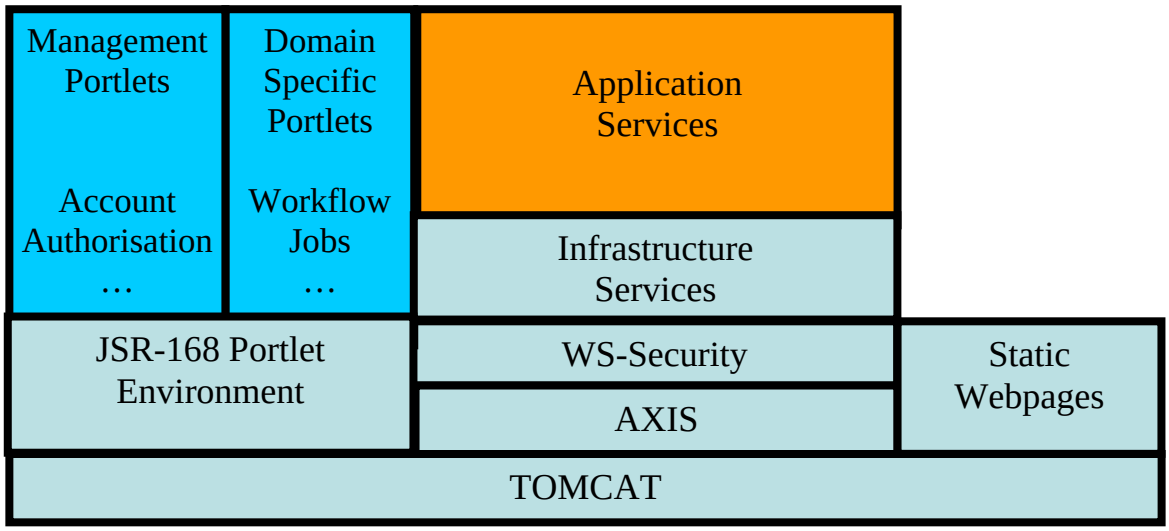


Figure 3 – The OMII Software Architecture

3.1 The OMII-UK Partnership

The creation of OMII-UK in January 2006 established a partnership between middleware activities at Southampton, Edinburgh and Manchester. This provides dedicated engineering support for:

- OGSA-DAI [OGSA-DAI] (Open Grid Services Architecture Data Access and Integration) is a middleware product which supports the exposure of data resources, such as relational or XML databases, onto Grids.
- TAVERNA [TAVERNA] The Taverna project aims to provide a language and software tools to facilitate easy use of workflow and distributed compute technology within the e-Science community.

3.2 Commissioned Software Development

The OMII at Southampton has allocated around half its budget to commissioning open-source software development within the community, which it runs on behalf of OMII-UK. The aim of this commissioning activity is the hardening of existing essential middleware components. These components provide additional functionality to the OMII software stack. The integration of the following components is currently in progress at the OMII: GridSAM (Job Submission & Monitoring service), BPEL (Workflow service), Grimoires (Registry service based on UDDI), FIRMS (Reliable messaging), FINS (Notification), GeodiseLab (Matlab toolbox), and the integration of WSRF::Lite into an Application Hosting Environment (AHE):

- GridSAM [GridSAM] is an open-source job submission and monitoring service. GridSAM installs on top of the WS-Security (authentication) layer provided by the OMII WS container and enables users to execute jobs on the OMII server that may have a variety of data input and output requirements. The GridSAM implements the Job Submission Description Language (JSDL) and is tracking the OGSA-BES (Basic Execution Service) from the Global Grid Forum.
- GRIMOIRES (Grid Registry with Metadata Oriented Interface: Robustness, Efficiency, Security) [GRIMOIRES] enables storage of service descriptions, distributed queries, WSDL documents and workflows. This registry also provides facilities for semantic annotation of information. Grimoires is fully UDDIv2 [UDDIv2] standard compliant. In addition to the UDDIv2 interface, Grimoires also provides some other interfaces, such as a metadata interface and a WSDL interface, which allow clients to publish and inquire over metadata and WSDL-related data, respectively. All the data published through various

interfaces are internally represented as RDF triples, which can be queried and reasoned about in a uniform way.

- FIRMS (Federation and Implementation of Reliable Messaging Specifications for Web Services) [FIRMS] represents an open source implementations of the WS-ReliableMessaging and WS-Reliability specifications.

- FINS (Federation and implementation of Notification Specifications for Web Services) [FINS] currently supplies open source implementations of the WS-Eventing specifications and later aWS-Notification implementation.

- BPEL (Business Process Execution Language) [BPEL] provides a flexible environment for the composition and enactment of e-science workflows using industry standard web service specifications.

- GeodiseLab [GeodiseLab] offers three toolboxes that provide facilities for accessing computing resources of various problem solving environments, data management, file transfer, and certificate handling.

- RAHWL (Robust Application Hosting with WSRF::Lite) [WSRFLite] – Builds upon a Perl implementation of WSRF family of specifications to provide simple lightweight clients to execute and control applications. This product provides support for several web service specifications such as: WS-Addressing, WS-ResourceProperties, WS-ResourceLifetimes, WS-BaseFaults, WS-ServiceGroups.

3.3 Integrated Services

The Integrated Services provides an Application Execution, Data Movement and Resource Allocation services that uses a common authorisation and business model to support the execution of pre-installed applications. The client command line tool provides the functionality to open accounts, to obtain resource allocations for computation and data use, manage access to these allocations, to upload input data/download output data and run applications pre-installed on server. An additional Account service is used to register with and manage access to the Integrated Services, and maintain account usage that records use against a defined quota. This service is being re-factored to support its use by services that are not part of the Integrated Service collection.

Two applications have been included in the latest software release that demonstrates how to use the Integrated Services - these are the OMII Test Application and Cauchy Horizons application. The functionality of OMII Test Application to check the correct installation and functionality of all components of the OMII software stack by providing a simple text sorting capability. Cauchy horizon is an application from the astro-physics community that calculates various parameters of space curvature in the vicinity of a black hole.

3.4 Authentication and Authorisation

OMII will continue to use X.509 certificates with the WS-Security framework to sign messages from both the client and the server. The X.509 certificates trust chain ends in a certificate from a recognised Certificate Authority. Mechanisms that move the storage of a user's long-lived certificate from their desktop(s) to a secure server will be explored to reduce the complexity of using X.509 certificates for the applied end-user.

An Authorisation service, based around the SAML (Security Assertion Markup Language) specification, will be integrated with the OMII WS Container to provide a single point of control to manage access to services (and eventually portals) hosted within the container. This infrastructure will form the basis for integration with national authorisation services.

4 Summary

There is no doubt that in case of the Grid we are dealing with a new phenomenon of unprecedented complexity that requires the solution, not only of technical problems, but of organisational and even political issues as well. Following on from the first wave of projects, which exploited new networking infrastructures, experimental test-beds, middleware and application software, we have a far greater understanding of the key issues. The next phase of activity will concentrate on interoperability and running applications that clearly

demonstrate the benefits of the Grid. There is a strong commitment to allocate more resources for this purpose that will certainly get us closer to the materialisation of the Grid promises.

Over the last two years OMII at Southampton, and now in partnership with Edinburgh and Manchester as the OMII-UK project, has become a source for reliable, interoperable and open-source Grid middleware components, services and tools to support advanced Grid-enabled solutions in academia and industry. The objectives of the OMII project are not to just develop the key components of a Grid infrastructure, but also to consolidate the expertise and intellectual capital invested in previous e-Science projects into well documented, robust and reliable software. Such a high-quality distributed software development process has rarely been attempted, or achieved, in the academic research community. By promoting the reuse of our software through documentation and support we believe we can enable our user community to spend more time on generating and evaluating ideas rather than getting lost in details of the technical work. The software repository being developed within OMII-UK is being extended within OMII-Europe to provide a general framework for assessing middleware for standards compliance, unit test coverage and other software metrics.

The coordination of eight commissioned software projects within the academic open-source development community also presents several challenges: the complexity of software, keeping pace with the fast moving area of Grid technology, interaction with the remote development and central integration staff, and the development of a coherent architecture across numerous development teams. At the OMII we have introduced policies (e.g. coding guidelines) to improve the software quality coming from the partners by defining templates and reviewing their functional specifications, design documents, implementation specification, testing plans, tutorials and user guides. The long-term aim of this strategy is to improve the efficiency of research projects, increasing the level of reuse between software projects and thereby to achieve a better utilisation of development resources.

References

[BPEL] http://www.ucl.ac.uk/research-computing/research/e-science/omii-bpel.html

[FINS] http://www.omii.ac.uk/mp/mp_fins.jsp

[FIRMS] http://www.omii.ac.uk/mp/mp_firms.jsp

[GeodiseLab] Geodise Project, http://www.geodise.org

[GridSAM] http://gridsam.sourceforge.net/2.0.0-SNAPSHOT/index.html

[GRIMOIRES] http://www.ecs.soton.ac.uk/research/projects/grimoires

[OGSA-DAI] OGSA-DAI Project, http://www.ogsadai.org.uk

[TAVERNA] Taverna component from the myGrid project. http://taverna.sourceforge.net/

[OMII] Open Middleware Infrastructure Institute, http://www.omii.ac.uk

[OMII-Europe] Open Middleware Infrastructure Institute Europe, http://www.omii-europe.com

[OMII-China] Open Middleware Infrastructure Institute China, http://www.omii -china.org/eng/index.htm

[UDDIv2] http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm

[WSRFLite] http://www.sve.man.ac.uk/Research/AtoZ/ILCT

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Job submission to grid computing environments

RP Bruin, TOH White, AM Walker, KF Austen, MT Dove
Department of Earth Sciences, University of Cambridge, Downing Street, Cambridge CB2 3EQ
RP Tyer, PA Couch, IT Todorov
CCLRC, Daresbury Laboratory, Warrington, Cheshire WA4 4AD
MO Blanchard
Royal Institution, 21 Albemarle Street, London W1S 4BS

Abstract

The problem of enabling scientist users to submit jobs to grid computing environments will eventually limit the usability of grids. The *e*Minerals project has tackled this problem by developing the "my_condor_submit" (MCS) tool, which provides a simple scriptable interface to Globus, a flexible interaction with the Storage Resource Broker, metascheduling with load balancing within a grid environment, and automatic metadata harvesting. This paper provides an overview of MCS together with some use cases. We also describe the use of MCS within parameter-sweep studies.

Introduction

For grid computing infrastructures to be exploited, it is essential that the tools built to provide access have usability designed into them from the outset. In our experience, it is unrealistic to ask most scientists to work with raw Globus job-submission commands – in the end they are likely to end up compromising by merely using `gsissh` to log into grid resources and submit jobs using more familiar batch queue commands. However, we have found that asking them to work with Condor job submission scripts is quite feasible [1]. In this paper we describe work we have done to develop Condor-like grid job submission tools that encompass integration with the San Diego Storage Resource Broker (SRB) [2] and new metadata capture tools [3].

The context for this work is the *e*Minerals minigrid structure [4,5]. This is a heterogeneous compute grid integrated with a data grid based on the SRB, as illustrated in Figure 1. Since the *e*Minerals project is an escience testbed project, we enforce a policy that access is controlled by the use of Globus job submission scripts (with authentication handled by Globus Grid Security Infrastructure (GSI) and X.509 digital certificates); we do not enable access via `gsissh` except for certain mission-critical exceptions (mostly for code developers and system administrators).

In previous papers we have described the *e*Minerals minigrid structure and access tools in some detail [4,5]. Our main job submission tool is "my_condor_submit" (MCS). This uses Condor-G, a Condor wrapping of Globus job-submission tools. As noted above, we quickly learned within the *e*Minerals project that users can work quite easily with Condor job submission scripts, and working with MCS only requires users to prepare a short Condor-like job submission script [1].

The first versions of MCS [4,5] essentially carried out one small set of tasks, namely to retrieve data from the SRB, run a program on the *e*Minerals minigrid, and then upload generated files to the SRB. Subsequent to this, we have developed a new version of MCS with some innovations that will be described in this paper. These include metascheduling, generalisation to other grid infrastructures (including campus grids and the National Grid Service), and automatic metadata capture.

MCS is primarily designed for users to submit one job at a time to a grid infrastructure from a computer on which the Globus toolkit has been installed. An issue that requires us to go beyond this approach is the need to wrap MCS within an infrastructure than enables the scientist to submit many jobs at a single instance as part of a combinatorial or ensemble study. We have developed scripts enabling automatic generation of input files for parameter

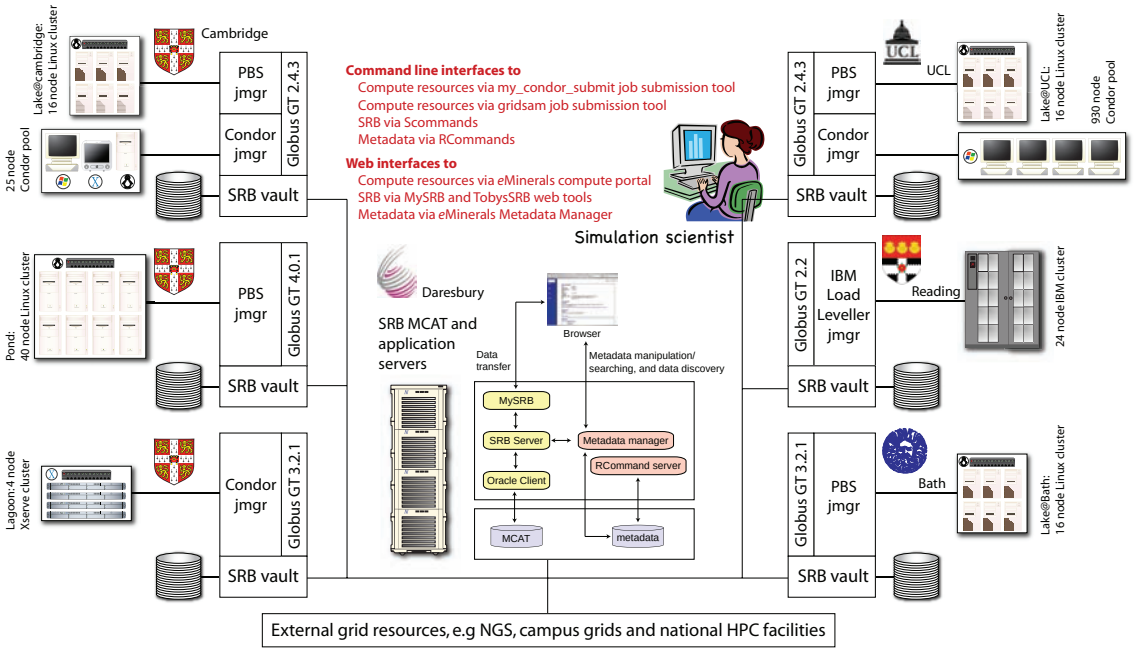


Figure 1. The eMinerals minigrid, showing the range of computational resources and the integrated data management infrastructure based on the use of the Storage Resource Broker.

sweeps which use MCS for their submission to address this need.

This paper describes how the *e*Minerals project handles grid computing job submission in a way that hides the underlying middleware from the users, and makes grid computing genuinely usable for scientists.

The *e*Minerals minigrid 2006

The current status of the *e*Minerals minigrid is illustrated in Figure 1. At the heart are two sets of components: a heterogeneous collection of compute resources, including clusters and Condor pools, and the data architecture based on the SRB with a central metadata catalogue installed onto a central application server, with distributed data vaults. Subsequent to our previous descriptions of the minigrid [4,5] are the following developments:

- Links to campus grids, e.g. CamGrid [6]. CamGrid is a heterogeneous Condor pool made up of several distributed and individually-maintained Condor pools which are integrated into a single grid through the use of Condor flocking. These resources are linked to the *e*Minerals minigrid by installing a Globus gatekeeper with a Condor jobmanager to allow job submission in the same manner as for other resources;
- Links to other grid resources such as the National Grid Service (NGS) and NW-grid [7] with seamless access, provided they have Globus gatekeepers and the possibility to install SRB and metadata tools;
- Creation of a metadata database and tools [3]. These enable automated metadata capture and storage so that simulation data can be archived in a manner that allows for much simpler retrieval at a later date. It also greatly facilitates collaborative working within a distributed virtual organisation such as *e*Minerals [8].

my_condor_submit 2006

MCS consists of a fairly complex Perl program and a central database, and is used with a simple input file with slight extensions over a standard Condor submission script. The primary purpose of MCS is to submit jobs to a grid infrastructure with data archiving and management handled by the SRB. The use of the SRB in our context serves two purposes. First, it is a convenient way to enable transfer of data between the user and a grid infrastructure, bypassing some of the problems associated with retrieval of multiple files whose names may not be known beforehand using Condor and gridftp methods for example. Second, the use of the SRB enables users to archive all files associated with a study in a way that facilitates good data management and enables collaborative access.

We have recently completely rewritten MCS. The original version [4,5] was written as a quick development to jumpstart access to the

*e*Minerals minigrid for the project scientists. It turned out that MCS was more successful than anticipated, mainly because it matched users' requirements and way of working. Because the original version was written quickly, it was increasingly difficult to upgrade and integrate new developments. Thus a complete rewrite proved to be essential to make future developments easy to implement.

The current version of MCS (version 1.2 as of July 2006) has the following features that extend versions previously described:

1. access to multiple collections (directories) on the SRB, since typically executables, data files and standard input files (such as pseudopotential files for quantum mechanical calculations) will be stored in different collections;
2. generalised to support job submission to external grid infrastructures, including campus grids and the NGS;
3. metascheduling with load balancing across all minigrid and external resources;
4. new metadata and XML tools to automatically obtain metadata from the output files and job submission parameters.

In the rest of this paper we will describe some of these in more detail.

MCS and metadata

We have been systematically enabling the simulation codes used within the *e*Minerals project to write output and configuration files in XML. Specifically, we use the Chemical Markup Language (CML) [9]. One author (TOHW) has developed a Fortran library, 'FoX', to facilitate writing CML files [9]. Another of the authors (PAC) has developed a library, called 'AgentX', to facilitate general reading of XML files into Fortran using logical mappings rather than a concrete document structure [10].

Our output XML files broadly have the data separated into three components; *metadata*, *parameter* and *property*. The *metadata* component consists of general information about the job, such as program name, version number etc. The *parameter* components are mainly reflections of the input parameters that control, and subsequently identify, the particular simulation. Examples range from the interesting parameters such as temperature and pressure to the more mundane but nevertheless important control parameters such as various cut-offs. The *property* components are the output data from the simulation. These lists are vast, and include step-by-step data as well as final averages or final values. Typically for metadata collection we need to retrieve some data from each of these three components. It is worth remarking with regard to the property metadata that we blur the distinction between data and metadata. This is illustrated by a simple example. In a study of the binding energy of a family of molecules (such as the dioxin family, where members of the family differ in the number and location of chlorine or hydrogen atoms), we store the data on final energy as metadata. This allows us to use this energy for our metadata search tools; an example would be to search through a study of all dioxin molecules for the molecule with the lowest binding energy.

In MCS, metadata is extracted from the XML data documents using the AgentX library. AgentX implements a simple API that can be used by other applications to find data represented in a document according to their context. The context is specified through a series of queries based on terms specified in an ontology. These terms relate to classes of entities of interest (for example '`Atom`', '`Crystal`' and '`Molecule`') and their properties (for example '`zCoordinate`'). This approach requires a set of mappings relating these terms to fragment identifiers. It is these identifiers that are evaluated to locate parts of documents representing data with a well defined context. In this way, AgentX hides the details of a particular data format from its users, so that the complexities of dealing with XML and the necessity of understanding the precise details of a particular data model are removed. The user is not required to understand the details of the ontology or mappings, but must have an understanding of the terms used.

Once data have been located using AgentX, MCS associates each data item with a term (such as '`FinalEnergy`') which is used to provide the context of the data. The data and term association are then stored in the project metadata database, making use of the RCommands [3].

We provide an example of extracting the final energy from a quantum mechanical energy relaxation using the SIESTA code [11]. In this work we use final energy as a metadata item because it is a useful quantity for data organisation and for search tools. The call to AgentX in MCS follows an XPath-like syntax:

```
AgentX = finalEnergy,
chlorobenzene.xml:/Module
[last]/PropertyList[title =
'Final Energy']/Property
[dictRef = 'siesta:Etot']
```

This directive specifies that MCS is to extract the metadata as a value from the file called '`chlorobenzene.XML`', and it will associate this value with the string called '`finalEnergy`'. The AgentX call looks for this value within the last `module` container, which by convention holds properties representing the final state of the system, then within a `propertyList` called '`Final Energy`', and finally within a `property` value defined by the dictionary reference '`siesta:Etot`'.

In addition to metadata harvested from output XML documents, we also collect metadata related to the user's submission environment. Examples include the date of submission, name of the machine submitted from, and the user's username on the submission machine. Metadata can also be collected about the job's execution environment, including the name of the machine on which the simulation was run, the completion date of the run, and the user's username on that machine. Users are also able to store arbitrary strings of metadata using a simple MCS command. All these types of metadata provide useful adjuncts to the scientific metadata harvested from the simulation output XML files.

MCS and metascheduling

One problem with the earlier versions of MCS was that the user had to specify which compute resource any simulation should be submitted to. This resulted in many jobs being submitted to a few busy resources within the *e*Minerals minigrid whilst other resources were left idle. Because users are not allowed to log in to resources, it was not possible for them to check job queues; in any case, such as approach is not a scalable solution to resource monitoring.

An intermediate solution to this problem was the creation of a status monitoring web page[1], which graphically shows the status of all available minigrid resources. However, this solution still requires user involvement to look at the page and decide where to run. Moreover, the web page caches the data for up to thirty minutes meaning that it would still be possible for users to submit to resources that have suddenly become busy since the last update.

To enable better load balancing across resources, querying of resource utilisation must be built into the submission mechanism, and must use up to date rather than cached data. This type of metascheduling, based on a round-robin algorithm, has now been built into MCS. To use this metascheduling facility, the user specifies the type of machine they wish to use, using the keywords '`performance`' to submit to clusters, or '`throughput`' to submit to Condor pools. MCS then retrieves a list of machines from a central database (which is mirrored for reliability), and queries the machines in turn checking for available processors. The database contains a list of available resources ranked in the order in which they were last submitted to (the most recently used machine appears last in the list) and other machine specific information such as the path to the required queue command and the machine's architecture, etc.

Querying of the resource state is performed by sending a Globus fork job to the resource that executes the relevant local queue command (obtained from the central database). Querying the status of a PBS cluster will result in the use of the '`pbsnodes`' command with suitable output parsing. Querying one of the available Condor pools will use a command written specifically for this purpose, wrapping the standard '`Condor_status -totals`'. The need for a purpose-written command is that this request will poll flocked Condor pools within a campus grid, and the standard queue queries do not return the required information in this case.

If all the resources are busy, MCS will inform the user and queue the jobs using information in the database to ensure even balance in the various resource queues.

As part of the metascheduling MCS takes account of the fact that not all binary exectuables will run on all resources. Again, this information is extracted from the database. Executables for various platforms are held within the SRB, and MCS will automatically select the appropriate SRB file download.

Example MCS files

Figures 2 and 3 show two examples of MCS input files. For those familiar with Condor, the Condor-G wrapping roots of MCS can be seen in the structure of the file. Figure 2 is the simplest example. The first three lines give information about the executable (name and location within the SRB) and the standard `GlobusRSL` command. The three lines with names beginning with `S` provide the interaction with the SRB. The `Sdir` line passes the name

[1] http://www.eminerals.org/gridStatus

```
# Specify the name of the executable to run
Executable      = gulp

# Specify where the executable should get stdin from and put stdout to
GlobusRSL = (stdin=andalusite.dat)(stdout=andalusite.out)

# Specify an SRB collection to get the relevant executable from
pathToExe        = /home/codes.eminerals/gulp/

# Specify a metadata dataset to create all metadata within
RDatasetId   = 55

# Specify a directory to get files from, put files to and relate to
# metadata created below
Sdir     = /home/user01.eminerals/gulpminerals/
Sget     = *
Sput     = *

# Creates and names a metadata data object
Rdesc    = "Gulp output from andalusite at ambient conditions"
# Specify metadata to get from files with Agent-x - get environment
# and default metadata only
AgentXDefault = andalusite.xml
GetEnvMetadata      = True
```

Figure 2: Example of a simple MCS script file.

of the SRB collection containing the files, and the "`Sput *`" and "`Sget *`" lines instruct MCS to download and upload all files. The lines beginning with `R` concern the interaction with the metadata database through the RCommands. The identification number of the relevant metadata dataset into which data objects are to be stored is passed by the `RDatasetID` parameter. The `Rdesc` command creates a data object with the specified name. Its associated URL within the SRB will be automatically created by MCS.

Figure 3 shows a more complex example, including the components of the simpler script of Figure 2. This script contains near the top parameters for the metascheduling task, including a list of specified resources to be used (`preferredmachineList`) and the type of job (`jobType`). The script in Figure 3 involves creation of a metadata dataset. It also contains commands to use AgentX to obtain metadata from the XML file. In this case, the study concerns an investigation of how the energy of a molecule held over a mineral surface varies with its *z* coordinate and the repeat distance in the *z* direction (`latticeVectorC`).

Parameter sweep code

Many of the problems tackled within the *e*Minerals project are combinatorial in nature. Thus a typical study may involve running many similar simulations concurrently (up to several hundreds), with each simulation corresponding to a different parameter value. Parameter sweep studies of this sort are well suited to the resources available in the *e*Minerals minigrid.

We have developed a set of script commands to make the task of setting up and running many concurrent jobs within a single study relatively easy. The user supplies a template of the simulation input file and information regarding the parameter values. The script then creates a set of collections in the SRB, one for each set of parameter values, containing the completed input files, and a commensurate set of directories on the user's local computer containing the generated MCS input files. The actual process of submitting the jobs is performed by running another script command, which then walks through each of the locally stored MCS input files and submits them all completely independently from each other.

Now having the tools for setting up, running, and storing the resultant data files for large numbers of jobs, the scientist then faces the problem of extracting the key information from the resultant deluge of data. Although the required information will differ from one study to another, there are certain commonalities in parameter sweep studies. The task is made easier by our use of XML to represent data. We have written a number of analysis tools for gathering data from many XML files, using AgentX, and generating XHTML files with

```
# Specify the executable to run
Executable      = siesta
# Instruct Condor to not tell us the outcome from the job by email
Notification     = NEVER

# Specify which file to use for stdin and stdout
GlobusRSL = (stdin=chlorobenzene.dat)(stdout=chlorobenzene.out)

# Force overwriting when uploading/downloading files
SForce     = true

# Specify an SRB collection to get the relevant executable from
pathToExe        = /home/codes.eminerals/siesta/
# Specify a list of machines that we are happy to submit to
preferredMachineList = lake.bath.ac.uk lake.esc.cam.ac.uk
lake.geol.ucl.ac.uk pond.esc.cam.ac.uk
# Specify the type of machine to be submitted to;
# “throughput: for a Condor pool and “performance” for a cluster
jobType         = performance
# Specify how many processors to use on the remote machine
numOfProcs       = 1

# Specify a metadata study to create a dataset within
RStudyId     = 1010
# Create and name a metadata dataset to contain data objects
RDatasetName   = "chlorobenzene on clay surface"

# Specify an SRB collection to do some transfers to/from
Sdir     = /home/user01.eminerals/clay_surface/
# Specify that we want to get every file from within this collection
Sget     = *

# Specify another SRB collection to do some transfers to/from
Sdir     = /home/user01.eminerals/chlorobenzene
# Specify that we want to put all local files into the specified collection
Sput     = *

# Create and names a metadata data object
Rdesc    = "chlorobenzene molecule on clay surface: first test"
# Specify metadata to get with Agent-x (Tied to the previous Sdir line)
# Get environment metadata
GetEnvMetadata = true
# Get default metadata from the specified file
AgentXDefault  = pcbprimfixed.xml
# Get z coordinate information and store as zCoordinate in the metadata
# database
AgentX     = zCoordinate, pcbprimfixed.XML:/molecule[1]/atom[last]/
zCoordinate
# Get lattice vector information and store in the metadata database
AgentX     = latticeVectorA, pcbprimfixed.xml:/Module/LatticeVector[1]
AgentX     = latticeVectorB, pcbprimfixed.xml:/Module/LatticeVector[2]
AgentX     = latticeVectorC, pcbprimfixed.xml:/Module/LatticeVector[3]
# Get the final energy from the file and store in the metadata database
AgentX     = finalEnergy, pcbprimfixed.xml:/Module[last]/PropertyList
[title='Final Energy']/Property[dictRef='siesta:Etot']
# Store an arbitrary string of metadata
MetadataString = arbString1, "First test of molecule height & z separation"

# Leave the code's stderr on the remote machine, to be uploaded to the SRB
# at job end
Transfer_Error = false

# End the file (taken from the Condor input file)
queue
```

Figure 3: Example of a more complex MCS input script

embedded SVG plots of interesting data [8,12].

To summarise, a full workflow can be constructed, starting with the parameter sweep information, creating multiple sets of input files, submitting and executing all the jobs, returning the data, and finally retrieving particular data of interest, and generating a single graph showing results from all jobs.

Case examples

In this section we describe some of the *e*Minerals science cases for which MCS has proved invaluable. Some, but not all, of these examples use the parameter sweep tools as well. All use the SRB, metascheduling and metadata tools within MCS. The key point from each example is that the process of job submission, including parameter sweep studies, is made a lot easier for the scientists.

Amorphous silica under pressure

Our work on modelling amorphous silica is described in detail elsewhere [13]. This study is easily broken down into many simulations, each with a different pressure but otherwise identical simulation parameters. The user provides a template input file with all of the necessary input parameters, and uses the parameter sweep tools to configure the scripts to vary pressure between the lower and upper values and in the desired number of steps (for example, values might vary from –5 GPa to +5 GPa in steps of 0.01 GPa, producing 101 simulations,). Once these values had been provided to the system then the user must simply run the command to set up the jobs and the data within the SRB, and then to submit the simulations to the available resources. MCS handles the complete job life cycle, including resource scheduling.

On completion of the jobs, the relevant task was to extract from all the resultant output XML files the values of pressure and volume. We also can monitor the performance and quality of the simulations by extracting XML data step by step.

Quantum mechanical calculations of molecular conformations

MCS was used extensively in the parameterisation of jobs to study the molecular conformations of polychlorinatedbiphenyl (PCB) molecules [14]. Preliminary calculations were required in order to determine the minimum space required to enclose a PCB molecule in vacuum by sampling over different repeat distances. The initial sweep involved a suite of 121 SIESTA calculations, but it subsequently transpired that more than one set of calculations was required. MCS facilitated an efficient use of the *e*Minerals compute resources, and led to the speedy retrieval of results. MCS also harvested metadata from the XML files, which was searchable by collaborators on the project, facilitating data sharing and results manipulation.

Pollutant arsenic ions within iron pyrite

In addition to the gains offered by MCS to combinatorial studies, the usefulness of MCS is not restricted to that kind of study; it can be generalised to all studies involving a large number of independent calculations. We highlight here an example treated in more detail elsewhere [15], namely the environmental problem of the incorporation in of arsenic in FeS_2 pyrite. This has been investigated by considering four incorporation mechanisms in two redox conditions through relatively simple chemical reactions. This leads to eight reactions with four to six components each. Therefore many independent calculations were performed for testing and obtaining the total energy of all reaction components using quantum mechanics codes. For this example, several submission procedures were needed, with at least one for each reaction component.

Summary and discussion

The problem of accessing grid resources without making either unreasonable demands of the scientist user or forcing users to compromise what they can achieve in their use of grid computing has been solved by the MCS tool described in this paper. MCS has undergone a complete reworking recently, and is now capable of metascheduling and load balancing, flexible interaction with the SRB, automatic metadata harvesting, and interaction with any grid infrastructure with a Globus gatekeeper (including Condor pools as well as clusters).

The only disadvantage faced by users of MCS is the need to have the Globus client tools and Condor installed on their desktop computers. With recent releases of Globus, this is less of a problem for users of Linux and Mac OS X computers, but remains a problem for users of the Windows operating system. A coupled problem may arise from firewalls with policies that restrict the opening of the necessary ports. For users who face either of these problems, our solution is to provide a group of submit machines from which users can submit their jobs using MCS. However, this is

less than ideal, so the next challenge we face is to bring MCS closer to the desktop. We are working on two solutions. One is the wrapping of MCS into a web portal, and the other is to wrap MCS into a web service.

Although MCS has been designed within the context of the *e*Minerals project, efforts have been made to enable more general usability. Thus, for example, it enables access to other grid resources, such as the NGS and NW-Grid [7], provided that they allow access via a Globus gatekeeper. The key point about MCS is that it provides integration with the SRB, and hence the SRB SCommand client tools need to be installed. To make use of the metadata tools (not a requirement), the RCommands and AgentX tools also need to be installed. These tools can be installed within user filespaces rather than necessarily being in public areas.

More details on MCS, including manual, program download, database details and parameters sweep tools, are available from reference 16.

Acknowledgements

We are grateful for funding from NERC (grant reference numbers NER/T/S/2001/00855, NE/C515698/1 and NE/C515704/1).

References

1. MT Dove, TO White, RP Bruin, MG Tucker, M Calleja, E Artacho, P Murray-Rust, RP Tyer, I Todorov, RJ Allan, K Kleese van Dam, W Smith, C Chapman, W Emmerich, A Marmier, SC Parker, GJ Lewis, SM Hasan, A Thandavan, V Alexandrov, M Blanchard, K Wright, CRA Catlow, Z Du, NH de Leeuw, M Alfredsson, GD Price, J Brodholt. eScience usability: the eMinerals experience. *Proceedings of All Hands 2005* (ISBN 1-904425-53-4), pp 30–37
2. RW Moore and C Baru. Virtualization services for data grids. in *Grid Computing: Making The Global Infrastructure a Reality*, (ed Berman F, Hey AJG and Fox G, John Wiley), Chapter 16 (2003)
3. RP Tyer, PA Couch, K Kleese van Dam, IT Todorov, RP Bruin, TOH White, AM Walker, KF Austen, MT Dove, MO Blanchard. Automatic metadata capture and grid computing. *Proceedings of All Hands 2006*
4. M Calleja, L Blanshard, R Bruin, C Chapman, A Thandavan, R Tyer, P Wilson, V Alexandrov, RJ Allen, J Brodholt, MT Dove, W Emmerich, K Kleese van Dam. Grid tool integration within the eMinerals project. P*roceedings of the UK e-Science All Hands Meeting 2004*, (ISBN 1-904425-21-6), pp 812–817
5. M Calleja, R Bruin, MG Tucker, MT Dove, R Tyer, L Blanshard, K Kleese van Dam, RJ Allan, C Chapman, W Emmerich, P Wilson, J Brodholt, A.Thandavan, VN Alexandrov. Collaborative grid infrastructure for molecular simulations: The eMinerals minigrid as a prototype integrated compute and data grid. *Mol. Simul*. **31**, 303–313, 2005
6. M Calleja, B Beckles, M Keegan, MA Hayes, A Parker, MT Dove. CamGrid: Experiences in constructing a university-wide, Condor-based grid at the University of Cambridge. *Proceedings of the UK e-Science All Hands Meeting 2004*, (ISBN 1-904425-21-6), pp 173–178
7. http://www.nw-grid.ac.uk/
8. MT Dove, E Artacho, TO White, RP Bruin, MG Tucker, P Murray-Rust, RJ Allan, K Kleese van Dam, W Smith, RP Tyer, I Todorov, W Emmerich, C Chapman, SC Parker, A Marmier, V Alexandrov, GJ Lewis, SM Hasan, A Thandavan, K Wright, CRA Catlow, M Blanchard, NH de Leeuw, Z Du, GD Price, J Brodholt, M Alfredsson. The *e*Minerals project: developing the concept of the virtual organisation to support collaborative work on molecular-scale environmental simulations. *Proceedings of All Hands 2005* (ISBN 1-904425-53-4), pp 1058–1065
9. TOH White, P Murray-Rust, PA Couch, RP Tyer, RP Bruin, MT Dove, IT Todorov, SC Parker. Development and Use of CML in the *e*Minerals project. *Proceedings of All Hands 2006*
10. PA Couch, P Sherwood, S Sufi, IT Todorov, RJ Allan, PJ Knowles, RP Bruin, MT Dove, P Murray-Rust. Towards data integration for computational chemistry. *Proceedings of All Hands 2005* (ISBN 1-904425-53-4), pp 426–432
11. JM Soler, E Artacho, JD Gale, A García, J Junquera, P Ordejón, D Sánchez-Portal. The SIESTA method for *ab initio* order-*N* materials simulation. J. Phys.: Cond. Matter **14**, 2745–2779, 2002
12. TOH White, RP Tyer, RP Bruin, MT Dove, KF Austen. A lightweight, scriptable, web-based frontend to the SRB. *Proceedings of All Hands 2006*
13. MT Dove, LA Sullivan, AM Walker, K Trachenko, RP Bruin, TOH White, P Murray-Rust, RP Tyer, PA Couch, IT Todorov, W Smith, K Kleese van Dam. Anatomy of a grid-enabled molecular simulation study: the compressibility of amorphous silica. *Proceedings of All Hands 2006*
14. KF Austen, TOH White, RP Bruin, MT Dove, E Artacho, RP Tyer. Using eScience to calibrate our tools: parameterisation of quantum mechanical calculations with grid technologies. *Proceedings of All Hands 2006*
15. Z Du, VN Alexandrov, M Alfredsson, KF Austen, ND Bennett, MO Blanchard, JP Brodholt, RP Bruin, CRA Catlow, C Chapman, DJ Cooke, TG Cooper, MT Dove, W Emmerich, SM Hasan, S Kerisit, NH de Leeuw, GJ Lewis, A Marmier, SC Parker, GD Price, W Smith, IT Todorov, R. Tyer , AM Walker, TOH White, K Wright. A virtual research organization enabled by eMinerals minigrid: An integrated study of the transport and immobilisation of arsenic species in the environment. *Proceedings of All Hands 2006*
16. http://www.eminerals.org/tools/mcs.html

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

The BROADEN Distributed Tool, Service and Data Architecture

Martyn Fletcher, Tom Jackson, Mark Jessop, Stefan Klinger, Bojian Liang, Jim Austin.

Advanced Computer Architectures Group,
Department of Computer Science,
University of York, YO10 5DD, UK.

Abstract

The organisation of much industrial and scientific work involves the geographically distributed utilisation of multiple tools, services and (increasingly) distributed data. A generic Distributed Tool, Service and Data Architecture is described together with its application to the aero-engine domain through the BROADEN project. A central issue in the work is the ability to provide a flexible platform where data intensive services may be added with little overhead from existing tool and service vendors. The paper explains the issues surrounding this and explains how the project is investigating the PMC method (developed in DAME) and the use of Enterprise Service Bus to over come the problems. The mapping of the generic architecture to the BROADEN application (visualisation tools and distributed data and services) is described together with future work.

1. Introduction

This paper describes an integrated and Distributed Tool, Service and Data Architecture developed for use in aero-engine health monitoring domain. The architecture has been developed to support the requirements of the BROADEN (Business Resource Optimization for Aftermarket and Design on Engineering Networks) project – part of the UK Department of Trade and Industry (DTI) Technology Innovation Programme. BROADEN is a follow on to the DAME (Distributed Aircraft Maintenance Environment) Grid pilot project [1].

Typically, as with many industrial applications and scientific research domains, the aero-engine domain needs to provide for distributed users with access to tools and distributed data. This coupled with Grid technologies [2] provide an opportunity to derive commercial and scientific benefit from distributed data. Such domains are often characterised by:

- Geographically distributed data – potentially in vast amounts.
- Geographically distributed users - not necessarily distributed to the same places as the data.
- Legacy Tools which are standalone but may need to interoperate with other tools.
- Tools which are designed to interoperate with other tools or services.
- Distributed services used by tools or other services.

A generic Distributed Tool, Service and Data Architecture is being developed to satisfy the above with aero-engine health monitoring as the exemplar domain. Modern aero engines operate in highly demanding operational environments with extremely high reliability. However, Data Systems & Solutions LLC and Rolls-Royce plc have shown that the adoption of advanced engine condition monitoring and diagnosis technology can reduce costs and flight delays through enhanced maintenance planning [3]. Such aspects are increasingly important to aircraft and engine suppliers where business models are based on Fleet Hour Agreements (FHA) and Total Care Packages (TCP). Rolls-Royce has collaborated with Oxford University in the development of an advanced on-wing monitoring system called QUICK [4]. QUICK performs analysis of data derived from continuous monitoring of broadband engine vibration for individual engines. Known conditions and situations can be determined automatically by QUICK and its associated Ground Support System (GSS). Less well-known conditions (e.g. very early manifestations of problems) require the assistance of remote experts (Maintenance Analysts and Domain Experts) to interpret and analyze the data. The remote expert may want to consider and review the current data, search and review historical data in detail and run various tools including simulations and signal processing tools in order to evaluate the situation. Without a supporting diagnostic

infrastructure, the process can be problematic because the data, services and experts are usually geographically dispersed and advanced technologies are required to manage, search and use the massive distributed data sets. Each aircraft flight can produce up to 1 Gigabyte of data per engine, which, when scaled to the fleet level, represents a collection rate of the order of Terabytes of data per year. The storage of this data also requires vast data repositories that may be distributed across many geographic and operational boundaries. The aero-engine scenario is also typical of many other domains, for example, many areas of scientific research, healthcare, etc.

We describe the development of an architecture which integrates geographically distributed users (tools), services and data. The following sections describe the generic Distributed Tool, Service and Data Architecture.

2. Architectural Issues and Development

This section provides an overview of the main characteristics of the tools, services, and data including the issues and rationale behind the development of the architecture.

2.1 Tool characteristics

A major issue within BROADEN has been the need to allow users to add tools to the system with the minimum of change to the tools. Our analysis has show the types of tools used can be characterised as follows:

- Tools designed to operate standalone; this is typical of legacy tools.
- Tools not designed to interoperate with other tools; again this is typical of legacy tools.
- Tools designed to use remote services.
- Tools designed to interact with other tools.

The main requirement is that the architecture should allow all the above tool types to interact with one another as necessary, with a minimum of change both to the tools and the system architecture (preferably none).

2.2 Service characteristics

Underlying the use of tools is the provision of services. These follow the requirements of the tools in section 2.1. The services used can be broken down into the following categories:

- Services may be centralised. This is typical in legacy systems.
- Service may be distributed – particularly located near the data sources.
- Service may be autonomous e.g. data input services triggered by data arrival.

The architecture should accommodate all the above services.

2.3 User characteristics

The system may have many geographically dispersed users, who may not be located in the same places as the data repositories. The architecture therefore should accommodate the use by geographically distributed users.

2.4 Distributed Data

The scenario, which we will describe, is one in which every time an aircraft lands, vibration and performance data is downloaded to the system from the QUICK system fitted to each engine. In future deployed systems, the volume of the Zmod data downloadable from the QUICK system may be up to 1 GB per engine per flight. Given the large volume of data and the rate at which it is produced, there is a need to consider how the data is managed.

In order to illustrate the requirements of the aero-engine application, consider the following scenario. Heathrow, with its two runways, is authorized to handle a maximum of 36 landings per hour [5]. Let us assume that on average half of the aircraft landing at Heathrow have four engines and the remaining half have two engines. In future, if each engine downloads around 1 GB of data per flight, the system at Heathrow must be capable of dealing with a typical throughput of around 100 GB of raw engine data per hour, all of which must be processed and stored. The data storage requirement alone for an operational day is, therefore, around 1 TB, with subsequent processing generating yet more data.

With such a vast amount of data being produced a centralised repository may place unacceptable demands on data bandwidth and will not provide a scaleable solution. Portions of data could be moved to services but this would represent only a solution in some limited application as most services will need the ability to use full data sets. Therefore, it is desirable to store data in distributed nodes and then the services which act on the data must also be distributed with the data – to avoid having to move the data to the services [6].

The architecture should permit the use of distributed nodes to store data (and services),

assuming other issues such as security, etc. are satisfied.

3. The Generic Distributed Tool, Service and Data Architecture

The scenario used for the DAME and BROADEN projects envisages geographically dispersed users using a workbench and tools which access distributed services and data.

Figure 1 provides an overview of the generic architecture that has been developed. The elements shown are:

- The Graphics and Visualisation Suites which contain a set of tools at a particular user location.
- The Enterprise Service Bus to enable tool interactions.
- The distributed nodes encapsulate the data and high performance services - see figure 2.
- The global workflow manager provides service orchestration on demand from the individual users or as an automatic activity in response to a predetermined stimulus.

Figure 2 shows the generic overview of the generic architecture:

- The Process Management Controller (PMC) is a distributed component which manages the distribution of service requests and collection and aggregation of results for return to the requester (tool).
- The Processing Services act on local data and provide high performance search, signal extraction facilities, etc.
- Distributed Data Management provides access to local and distributed data through Storage Request Broker abstraction mechanisms.
- The Local Workflow Manager Management provides automatic and requested workflows to be enacted with a single distributed node. A local workflow may also be part of a global workflow controlled by the global workflow manager.
- The Local Resource Broker manages selection of local resources in keeping with specified Service Level Agreements (SLAs).
- Data Loading Services populate the local data stores in each node. These services may also perform other tasks such as data integrity checking, error correction, etc. on input data.
- Databases / Data stores are provided for each type of data resident in the node.

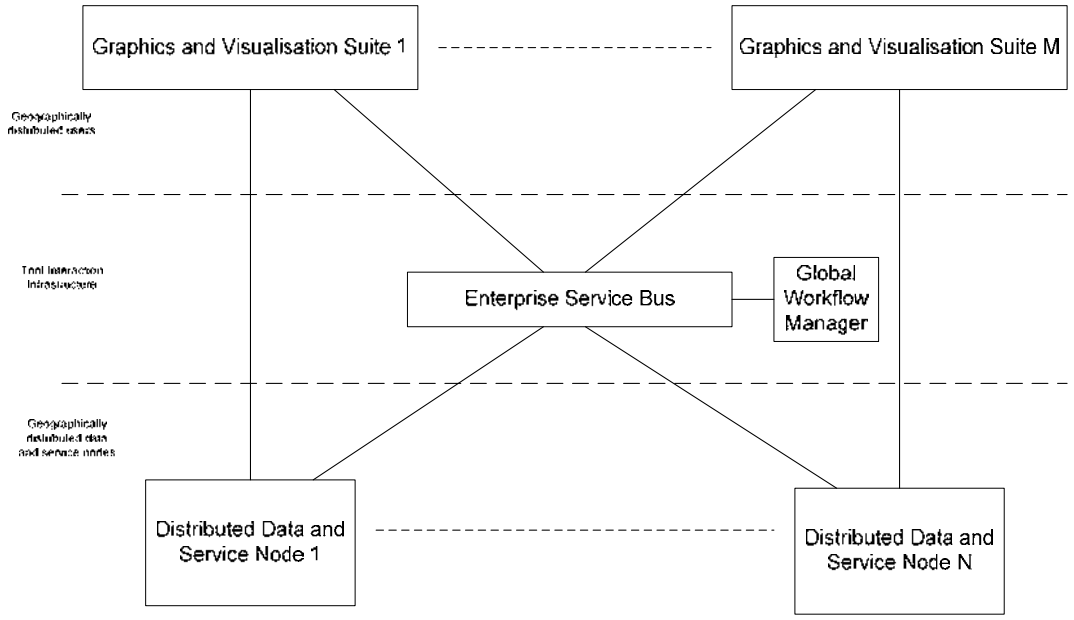


Figure 1 Overview of the Generic Tool, Service and Data Architecture

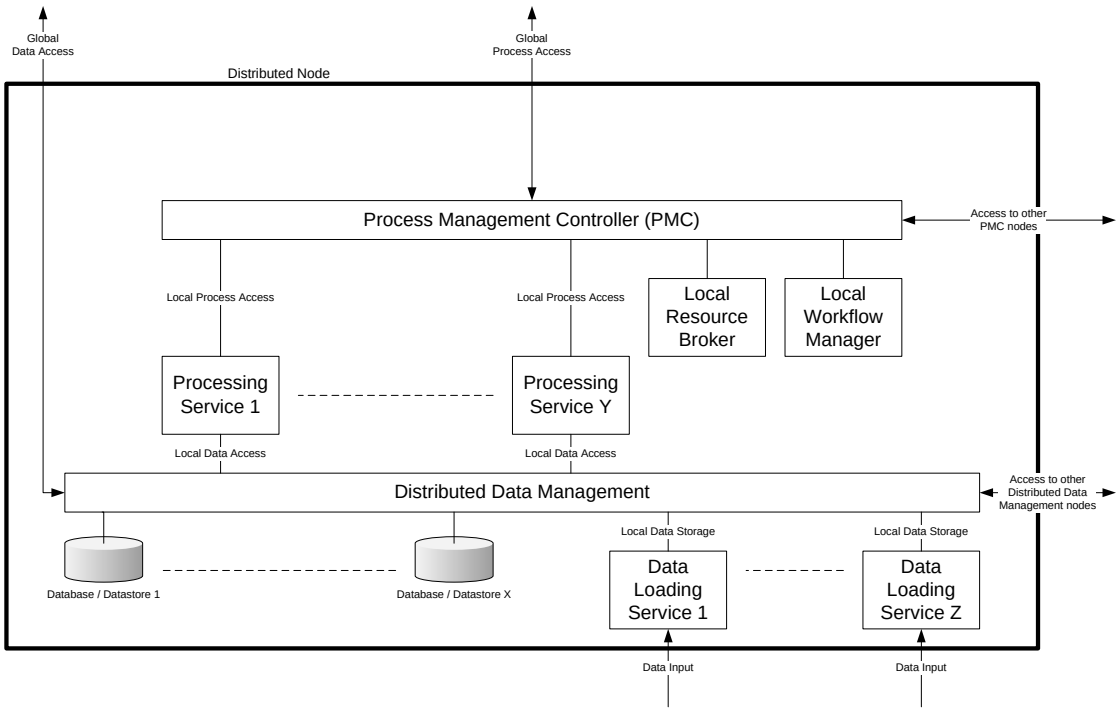


Figure 2 Overview of a Generic Distributed Service and Data Node

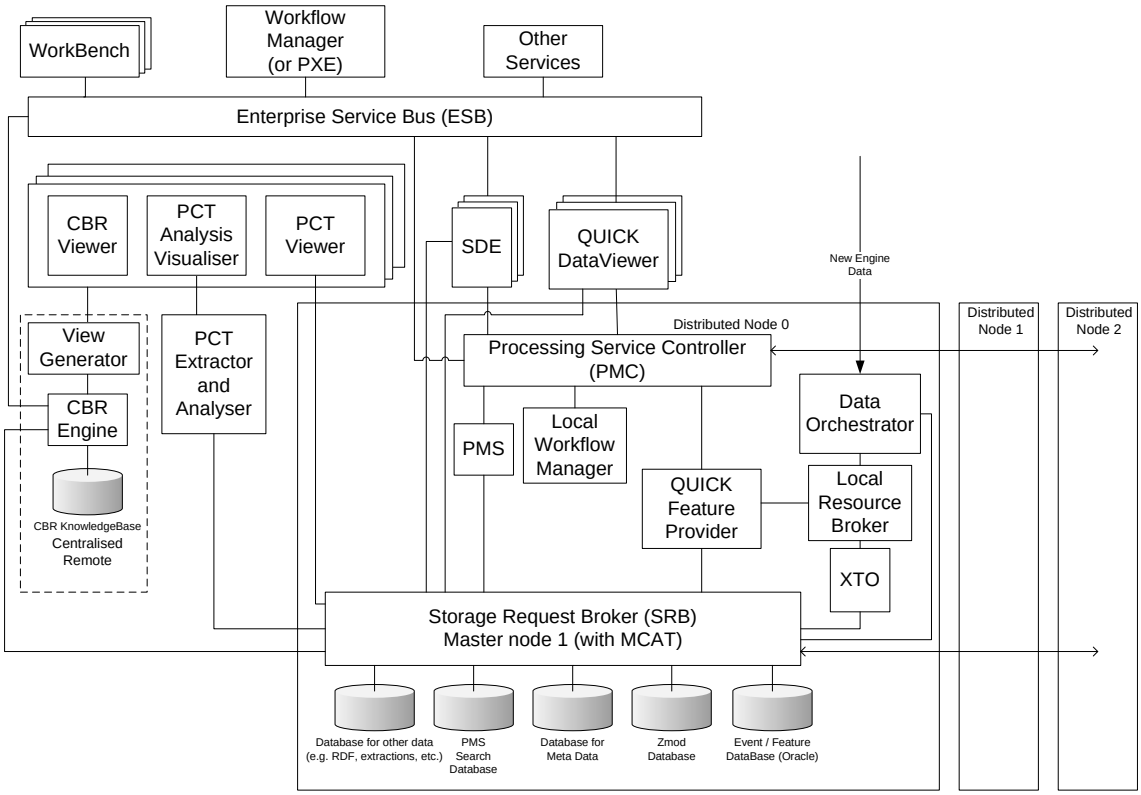


Figure 3 Overview of BROADEN Tool, Service and Data Architecture

4. Tool Interaction and Distributed Service and Data Management

This section describes the main middleware elements we have selected which allow the tools to interact and which facilite the management of distributed data and services. As shown in Figure 3, there are a number of tools that communicate with each other. These originate from different vendors (in our case Oxford Biosignals, York University, Sheffield University, Leeds University and Rolls-Royce). There are many other tools and services that could be added from other suppliers. The main challenge has been to allow these tools and services to communicate with each other without having to change the tools or services themselves, that is, to provide an open and extensible architecture. This is driven by the recognition that tool and services vendors are unlikely to want to alter their software just to fit into the proposed architecture, or to communicate with a new tool (in the project we have simulated this constraint). The simplest mechanism to solve this problem would be to implement adapters between all the services and tools that translate the communications between the tools. Unfortunately with a large number of tools and services the number of adapters would potentially be very large. This raises the need for a middleware layer to allow a configurable translation mechanism. This middleware must also address work orchestration issues such as starting, stopping, monitoring and dynamic workflow capabilities. Our work has investigated the Enterprise Service bus (ESB) and our own PMC to achieve this.

4.1 Enterprise Service Bus

An Enterprise Service Bus [7] typically provides XML messaging, XML transformation, intelligent routing, connectivity, support for service oriented architectures, etc. An instance of an ESB may be used within the architecture as a simple messaging and translation mechanism between the tools of the system. Tools are able to register with the ESB providing information about their type of service, their functions and the respective data formats. Translation specifications are provided to the ESB in a standard format and the ESB provides translation facilities on a tool pair basis and even on a function pair basis, if necessary.

The ESB keeps a registry of all connected tools, and routes messages between tools and translator components. Tools, once registered, might become unavailable, move to different locations or change their data formats without informing the ESB. Therefore a method of continuous verification and notification of each service will also be implemented.

Errors and exceptions are logged for an administrator to resolve problems. In addition to this, an informative message is returned to the client with information about the failure of any requests.

Workflow, for use in the Global Workflow Manager, can be expressed in BPEL4WS and in order to make the update of workflows more user-friendly, a GUI will be included in the Workbench. Our concerns with ESB are performance related, given its requirement to communicate primarily through XML schemas.

4.2 Process Management Controller

We have developed a complementary technology in DAME, which is termed Process Management Controller (PMC). PMC is a lightweight middleware application for managing services which act on distributed data in a generic manner and for providing high performance communication [6]. The PMC is independent of the data types and algorithms used and provided by the services.

The PMC service provides an interface that allows client applications (tools or other services) to manage services and retrieve results from a network of PMC nodes, without knowledge of how many nodes exist or where they are located. Each service operates on data held within its local node.

The network of PMC nodes is organised in a peer-to-peer network of independent nodes. When a distributed service activity is initiated at a PMC node, this node becomes the master for that activity. All other nodes are designated slaves for that activity. The master node replicates the service request to all the slave nodes. Each node then uses the appropriate local service to perform the activity and the slave nodes return their results to the master node, where they can be collected by the client application. Subsequent or parallel service activities can be initiated at any node in the network.

On start-up each PMC node reads its local configuration file which contains parameters that adequately describe its state, including its *seed* PMC node. The seed node is a key aspect to the PMC network initialisation process. In an empty network, one initial node is designated the seed node, and this node is started first. All

subsequent nodes can use this node as their seed. At later points in time, it may be desirable to use additional seed nodes. The initialisation process for both new and existing nodes will be synchronised and measurable. The PMC nodes will form a peer-to-peer network where the nodes are loosely coupled and only form connections in response to user requests.

As new nodes are added and they register with all other nodes in the network, each node will maintain a persistent register of known nodes.

It may also be necessary to temporarily remove nodes from the network e.g. for maintenance. In these cases the node will not be logically removed, so as to maintain the data coverage statistics for search operations. When these nodes are restarted, they will need to ensure that their node lists are correct and up to date. If a node is to be permanently removed from the PMC network, then it must use the de-register method on each PMC in its contact list.

4.3 PMC v ESB

We currently see the capabilities of PMC and ESB being complementary; one offers flexibility at the cost of performance, the other reduces the flexibility but give a gain in performance. An analysis is underway to measure these issues within the BROADEN application. We see the eventual possibility of migrating the functional aspects of ESB that we determine as essential into PMC, once these have become clear through trial deployment.

4.4 Distributed Data Management.

As the data within the system may be distributed between many compute nodes it is necessary to provide a mechanism to virtualise the storage across the nodes. Our current choice is the SDSC Storage Request Broker (SRB) [8]. SRB is a tool for managing distributed storage resources from large disc arrays to tape backup systems. Files are entered into SRB and can then be referenced by logical file handles that require no knowledge of where the file physically exists. A metadata catalogue is maintained which maps logical handles to physical file locations. Additional system specified metadata can be added to each record.

The SRB can operate in heterogeneous environments and in many different configurations from completely stand-alone, such as one disc resource, one SRB server, and one MCAT, to completely distribute with many resources, many SRB servers, and completely federated MCATs. In this configuration, a user could query their local SRB system and yet work with files that are hosted remotely.

When data is requested from storage, it is delivered via parallel IP streams, to maximize network throughput, using protocols provided by SRB. In current implementations, a single MCAT is deployed inside a Postgress database. Real world implementations would use a production standard database, such as Oracle or DB2. With this setup, each nodes's SRB installation is statically configured to use the known MCAT.

The use of a single MCAT provides a single point of failure that would incapacitate the entire distributed system. To alleviate this, a federated MCAT would be used, allowing each node to use their own local MCAT. In large systems, this would provide a performance benefit, as all data requests would be made through local servers. However, this would require an efficient MCAT synchronization process.

5. The BROADEN Tool, Service and Data Architecture Implementation

The previous two sections described the generic architecture. This section describes the actual tools and services used when the architecture is applied to the BROADEN project. Figure 3 shows the architecture populated with the BROADEN tools and services.

The visualisation tools provided are:

- Case Based Reasoning Viewer [9] including Performance Curve Test (PCT) Viewers.
- Signal Data Explorer (SDE) is a visualisation tool used to view vibration spectral and extracted time series data [5, 10] and performance data. It also acts as the front end to the high performance pattern matching services.
- QUICK Data Viewer is a visualisation tool which allows a user to view spectral data, extracted features and events occurred [4].

The data repository at each node is responsible for storing all raw engine data along with any extracted and AURA encoded data. Each node is populated with:

- The Pattern Match Service (PMS) which is based on Advanced Uncertain Reasoning Architecture technology [11] and provides high performance pattern matching for use with the SDE or in workflows.

- The QUICK Feature Provider service which provides features and events to a client such as the QUICK Data Viewer.
- The XTO (eXtract Tracked Order) service which extract specific time series from the raw spectral vibration data.
- The Data Orchestrator is the Data Loading Service and is responsible for "cleaning" and storing all raw engine data.

Also included in the BROADEN architecture is a centralised CBR Engine and Knowledge Base [9].

6. Future work

The generic architecture has been developed to be application neutral. It will be implemented for the BROADEN application during the course of the project. This will allow us to assess the strengths and weaknesses of PMC and ESB for the task.

Future work will include:

- The testing and demonstration of the generic architecture in the BROADEN application domain.
- The testing and demonstration of the generic architecture in other application domains.
- The introduction of fault tolerance as appropriate (inter and intra node).
- The exploration of fault tolerance techniques within the ESB.
- Performance testing would be helpful to discover issues early in the process.

7. Conclusions

This work builds on the tools and services developed during the DAME project. A generic architecture has been developed, which integrates:

- Geographically distributed nodes containing data repositories and services.
- Geographically distributed users.
- Legacy Tools.
- Purpose designed tools.

It is a Grid based architecture used to manage the vast, distributed, data repositories of aero-engine health-monitoring data. In this paper we have focused on the use of a generic architecture to enable the general concept to be used in other application areas and with varying degrees of legacy systems and designs.

The middleware elements are generic in nature and can be widely deployed in any application domain requiring distributed tools and services and data repositories.

8. Acknowledgements

The work reported in this paper was developed and undertaken as part of the BROADEN (Business Resource Optimization for Aftermarket and Design on Engineering Networks) project, a follow-on project to the DAME project. The BROADEN project is part-funded via the UK Department of Trade and Industry under its Technology Innovation Programme and the work described was undertaken by teams at the Universities of York, Leeds and Sheffield with industrial partners Rolls-Royce, EDS, Oxford BioSignals and Cybula Ltd.

9. References

[1] J. Austin et al., "Predictive maintenance: Distributed aircraft engine diagnostics," in The Grid, 2nd ed, I. Foster and C. Kesselman, Eds. San Mateo, CA: Morgan Kaufmann, 2003, Ch. 5.

[2] I. Foster and C. Kesselman, Eds., The Grid, 2nd ed. San Mateo, CA: Morgan Kaufmann, 2003.

[3] Data Systems and Solutions Core Control™ technology. http://www.ds-s.com/corecontrol.asp.

[4] A. Nairac, N. Townsend, R. Carr, S. King, P. Cowley, and L. Tarassenko, "A system for the analysis of jet engine vibration data," Integrated Computer-Aided Eng., vol. 6, pp. 53–65, 1999.

[5] D. O'Connel, "Pilots predict air chaos," The Sunday Times, p. 3.3, Mar. 14, 2004.

[6] J. Austin et al., "DAME: Searching Large Data Sets Within a Grid-Enabled Engineering Application". Proceedings of the IEEE, vol. 93, no. 3, March 2005.

[7] David A. Chappell. "Enterprise Service Bus – Theory in Practice". O'Reilly. ISBN 0-596-00675-6.

[8] SDSC Storage Request Broker [Online]. Available: http://www.sdsc.edu/srb/

[9] M. Ong, X. Ren, G. Allan, V. Kadirkamanathan, H. A. Thompson, P. J. Fleming (2004). "Decision support system on the Grid". Proc Int'l Conference on Knowledge-Based Intelligent Information & Engineering Systems, KES 2004.

[10] Martyn Fletcher, Tom Jackson, Mark Jessop, Bojian Liang, and Jim Austin. "The Signal Data Explorer: A High Performance Grid based Signal Search Tool for use in Distributed Diagnostic Applications." CCGrid 2006 – 6th IEEE International Symposium on Cluster Computing and the Grid. 16-19, May 2006, Singapore.

[11] The AURA and AURA-G web site, Advanced Computer Architectures Group, University of York, UK. http://www.cs.york.ac.uk/arch/NeuralNetworks/AURA/aura.html.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

gridMonSteer: Generic Architecture for Monitoring and Steering Legacy Applications in Grid Environments

Ian Wang[1,2], Ian Taylor[2,3], Tom Goodale[2,3], Andrew Harrison[2] and Matthew Shields[1,2]
[1]School of Physics and Astronomy, Cardiff University
[2]School of Computer Science, Cardiff University
[3]Center for Computation and Technology, Louisiana State University

Abstract

Grid environments present a virtual black box to scientists running legacy applications, with the Grid infrastructure effectively hiding the running application on a resource over which the scientist generally has limited or no control. Existing monitoring systems that allow Grid-enabled applications to communicate their progress and receive steering information are inapplicable as they require code modification, and this not possible in true legacy scenarios. While a black box approach may be acceptable in batch execution scenarios, it means the Grid is cut off to legacy applications where interactions or intermediate results are required.

In this paper we present gridMonSteer, a simple, non-intrusive architecture that allows scientists to receive intermediate results and interact with legacy applications running in a Grid environment. This architecture is Grid middleware independent and can be employed to monitor applications submitted via any Grid resource manager (e.g. GRAM, GRMS or Condor/G) or running within a Grid service framework. We present a case study describing how gridMonSteer enables legacy applications to act as active components in Grid workflows, dynamically driving and steering the workflow execution.

1 Introduction

Within a Grid environment, the vast majority of data analysis applications executed by scientists can be considered legacy. By this we mean that they are unaware of their Grid environment and any mechanisms it provides to communicate with users or controller applications. Although systems exist that facilitate communication with Grid applications [1][2][3], there is a general reluctance to re-engineer or rewrite legacy applications to utilize any communication mechanisms available due to the cost and lack of an agreed standard. For scientists using legacy applications the Grid environment acts as virtual black box, in which jobs are submitted and executed but retrieving intermediate results or interacting with the running application is very difficult.

Opening up the Grid environment to interactive legacy applications will allow us to much better integrate these applications into distributed problem solving environments. Rather than just viewing legacy application as standalone entities, we can now employ them as active components in larger complex decision-based applications. For example, the intermediate results generated by a legacy application can be used to dynamically steer a data-driven Grid workflow.

In this paper we present gridMonSteer, a generic architecture for monitoring and steering applications in a Grid environment, together with its current implementation. This implementation provides many monitoring and steering capabilities both at the application level and in terms of a dynamic Grid workflow. The principal benefit of gridMonSteer is that simple, often generic solutions to Grid application monitoring/steering scenarios can be developed without modifying any application code. These solutions are Grid middleware independent and function equally whether the application is submitted using GRAM [4], GRMS [5] or Condor/G [6], or even run within a Grid service

framework [7].

The gridMonSteer architecture consists of two parts: an application wrapper that runs in the same execution environment as the application, and an application controller that generally runs locally to the user. As all the communication in gridMonSteer is initiated by the application wrapper, all communication is outbound from the Grid resource. This situation is generally allowed by firewalls as it implies that the application wrapper has already complied with the security requirements of that resource.

The gridMonSteer application wrapper is executed as part of the application job submission. The arguments specified in this submission tell the wrapper what information to request from/notify to the gridMonSteer controller, and these can be tailored to a specific application scenario. This information could for example be immediate notification of output files created by the application, incremental requests for input to the application (e.g. via the standard input), requests for input to be forwarded to the application via a network port, or notification of the state of the application.

A gridMonSteer application controller is a service generally run locally by the user that receives information requests and notifications from the application wrapper. A controller exposes a simple interface that can be invoked by the application wrapper; in the current implementation this is web service interface. As long as this interface is exposed, the actual behavior of controller is undefined and can be tailored to individual or generic application requirements; for example:

- a generic controller could combine incremental input to the standard input with immediate output from the standard output to conduct an interactive command-line session with a Grid application.
- a visualization controller could receive immediate notification of image files to produce a live animation of output from a Grid application.
- a workflow controller could use immediate output file notification from an application to dynamically steer the execution of a Grid workflow, e.g. to launch an immediate response to a significant event.

The rest of this paper further describes the architecture, implementation and application of gridMonSteer. In Sections 3 and 4 respectively we explain the gridMonSteer architecture and our implementation of this architecture. We present a case study in Section 5 that uses gridMonSteer in combination with Triana [8][9] and Cactus [10][11] to dynamically steer a Grid workflow. We draw our conclusions from this work in Section 6. First however, in Section 2, we look at work related to the architecture presented in this paper.

2 Related Work

The number of scientific applications developed to operate in a Grid environment is tiny when compared with the vast number that can be considered to be legacy codes when run within such environments. Grid legacy applications are not implemented with hooks into the application monitoring systems that are being developed for Grid environments, such as Mercury [2] and OMIS [3] (see [1] for a survery), and therefore have no method for communicating their progress/intermediate results or receiving steering information.

One obvious solution to this problem is to "Grid-enable" these applications by allowing them to interact with a monitoring system. However, enabling an application to interact with existing monitoring systems requires modification of the application source code. In a legacy application scenario this is often not possible because the source code is unavailable. If the source code is available, the difficulty and cost of modifying unsupported code is likely to prove prohibitive, especially when standards for monitoring architectures in Grid environments are still being developed.

Without access to monitoring systems, when a legacy application is submitted to execute on the Grid via a resource manager, such as GRAM, GRMS or Condor/G, that resource manager becomes the principal point of contact regarding the progress of that application. However, the progress information available to a user is generally extremely limited, not normally stretching much beyond the status of the job and its id. Although a user could expect to have access to the execution directory of the legacy application, this access is pull based, so retrieving intermediate results in

a timely fashion would require constant polling of the remote resource. This is very inefficient compared to the result notification approach used in gridMonSteer and unlikely to prove a usable architecture for effective application controllers.

An alternative to using a resource manager to execute a legacy application is to expose the functionality of the application as a Web or Grid service. To achieve this requires an application wrapper that mediates between the service interface and invocations of the application. A couple of approaches, including SWIG [12] and JACAW [13], have adopted a fine-grained approach where the actual source code is wrapped exposing the application interface. However, in addition to granularity and complexity considerations, this approach requires access to the application source code, a situation that is generally not applicable to legacy applications.

A coarse grained approach, where the application is treated as a black box interfaced with using stdin/stdout or file input/output, is used in other projects, such as GEMLCA [14] and SOAPLab [15]. This approach however typically provides little additional benefit over using a resource manager with regards to application interaction. This is illustrated by the fact that, rather than wrapping the application process directly, GEMLCA actually wraps a Condor job submission.

One system that does employ similar components to that of gridMonSteer is Nimrod/G [16][17], a Grid based parametric modeling system. In Nimrod/G an application wrapper is used to repeatedly to run parametric modeling experiments with varying parameters and to pre/post stage results. A central database is queried by the application wrapper to determine the next experiment to execute. Such a scenario could easily be implemented using the dynamic scripting capacity of the current gridMonSteer implementation (see Section 4.1.1), with the central database acting as the application controller. Unlike gridMonSteer, Nimrod/G does not provide a generic legacy application monitoring or steering architecture outside dynamic scripting of parameter modeling experiments.

3 gridMonSteer Architecture

The gridMonSteer architecture (see Figure 3) consists of two parts, an application wrapper that is executed on a Grid resource and an application controller that is generally run locally to the user. As gridMonSteer application wrapper is submitted to the Grid as a standard job, this architecture works with any Grid resource manager (e.g. GRAM, GRMS or Condor/G). Unlike other Grid monitoring approaches, the gridMonSteer application wrapper is non-intrusive and therefore does not require modifications to the application code.

The process of running an application under gridMonSteer is initialized when, rather than submitting an application to a Grid resource manager for execution, a gridMonSteer application wrapper job is submitted instead (Step 1 in Figure 3). The original application executable and arguments are supplied as arguments to the wrapper job. As the wrapper job is submitted to the resource manager as a standard job, all the features of the chosen resource managers job description are available, as they would be if submitting the application directly.

As well as the application executable and arguments, the address of a gridMonSteer application controller service is also supplied as an argument to the wrapper job. An application controller is a service (e.g. web service) that exposes a simple network interface (see Section 4.2). This interface is invoked by the application wrapper to request information (e.g. input files) from and notify information (e.g. output files) to the controller. Other arguments supplied to the wrapper job specify exactly which information is notified to/requested from the controller by the application wrapper, as will be discussed in Section 4.1.

The gridMonSteer application wrapper job is executed on a Grid resource as specified in the job description and determined by the resource manager (Step 2). The application wrapper is then responsible for launching the original application (Step 4), however, before this is done, any input files that are required to be pre-staged in the execution directory are requested from the controller by the application wrapper (Step 3).

After the application has been launched by the application wrapper, the application wrapper regularly monitors the execution directory for the creation/modification of output files (Step 5). If any changes occur to files required by the application controller, as specified in the wrapper job arguments (see Section 4.1), then the controller is noti-

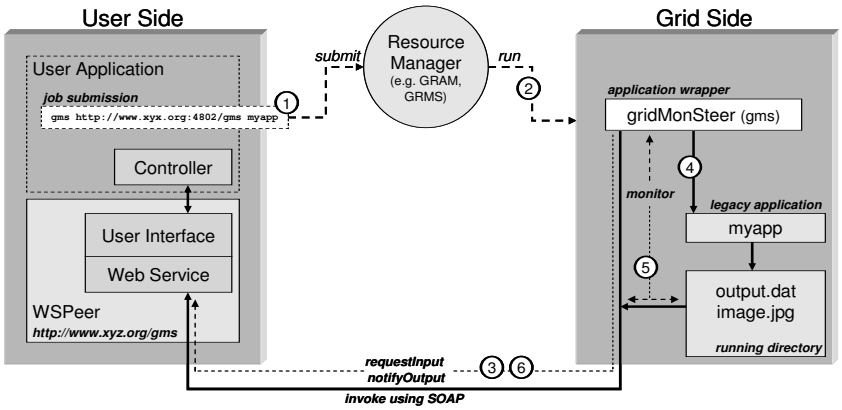


Figure 1: gridMonSteer architecture using a web service controller launched using WSPeer.

fied via its network interface (Step 6).

In addition to handling input/output files, the application wrapper can perform other notification/steering tasks such as job state notification and dynamic script execution. We detail such functionality further in Section 4.

As long the application controller exposes the predefined controller interface, the gridMonSteer architecture does not define how the controller handles the information requests/notifications that it receives from the wrapper.

3.1 Security Considerations

In a Grid environment most hosts are behind firewalls that typically create a one-way barrier for incoming traffic on specified ports, thereby disabling incoming connections from external controllers. Hosts can also employ address translation that is not Internet facing and therefore cannot be contacted directly, such as Network Address Translation (NAT) systems. Once this job is started in such an environment direct communication from an application controller becomes difficult or impossible.

In the gridMonSteer architecture all communication is initiated by the gridMonSteer application wrapper. As the application wrapper executes in the same environment as the Grid application this means that communication is always outbound from Grid resources. This is a situation that is generally allowed by firewalls as it implies the application wrapper has already complied with the security requirements of that resource. In the case of gridMonSteer, as the application wrapper is submitted as a standard Grid job, these security requirements are enforced by the resource manager used to execute the wrapper job. Outbound communication is also compatible with address translation systems such as NAT.

As long as a secure transport protocol such as HTTPS is used for communication between the application wrapper and controller, identity certificates can be used by the application wrapper to verify it is communicating with a trusted controller. Further from that, it is up to the application controller implementation not to introduce security flaws. The application controller should operate under the same trust relationship that allowed the application wrapper job to be executed on the Grid resource initially.

4 gridMonSteer Implementation

The designs of the current gridMonSteer application wrapper and application controller implementations are described in Sections 4.1 and 4.2 respectively. At present the application wrapper is implemented in Java, however a C version is under development. In the current implementation the controller exposes a web service interface, and can be implemented in any language so long as this

requirement is met.

4.1 Application Wrapper

As described in Section 3, in the gridMonSteer architecture, rather than submitting an application directly to a Grid resource manager, an application wrapper job is submitted. This application wrapper is then responsible for requesting input information from the application controller, launching the original application, monitoring the output from this application, and notifying this information to the application controller.

When the application wrapper job is submitted the following arguments must be specified:

- The address of the application controller, which in the current implementation is the address of a web service.
- The application executable and any arguments that should be passed to this application.

Additional arguments can then be used specify what information is requested from and notified to the application controller based on the individual application scenario. We outline these arguments in Sections 4.1.1 and 4.1.2.

4.1.1 Monitoring Arguments

The set of gridMonSteer arguments that we refer to as monitoring arguments are used to specify the information that is sent from the application wrapper (grid side) to the controller (user side). The following arguments are used to specify which files output by the legacy application are notified to the controller, and the notification policy used:

`-out` - Sends output files to the controller after application execution has finished.

`-monitor` - Same as `-out` except the output files are sent immediately after their creation/modification.

`-update` - Same as `-out` except incremental updates to the output files are sent periodically during application execution.

Each of the arguments above take a file list which details the names of files and/or directories to be monitored, and may include wildcards (e.g. *.jpg). The keywords `STDOUT` and `STDERR` can be used in the file list to specify monitoring the standard output and standard error streams respectively.

Although the basic file notification arguments detailed above covers most monitoring situations, in some scenarios applications generate a large number of output files of which the scientist is only interested in a subset determined during application execution. For these scenarios gridMonSteer uses a register/select system, whereby output files are registered with the application controller when they become available and the application wrapper requests the controller/user to select the files that are of interest. The `-xout`, `-xmonitor` and `-xupdate` arguments provide register/select compliments for the basic file notification arguments.

In addition to output file notification, the `-state` argument can be used to indicate that application state information, such as host name, running directory and run time, should be notified to the application controller periodically during execution. Such information is particularly useful in a grid environment where the application may reside in a job queue for some time and execute on an unfamiliar, remote resource.

4.1.2 Steering Arguments

The gridMonSteer steering arguments specify the information that is requested from the application controller (user side) by the application wrapper (grid side). The `-run` argument specifies that the wrapper asks the whether the application should be re-run after it has exited. On each run the job arguments or even the job executable can be altered. This effectively allows the controller to execute a dynamic script via the application wrapper. This offers considerable benefits over the alternative, which would be to submit a succession of independent jobs, such as not incurring multiple scheduling delays and not having to deal with the complexity of the independent jobs being run in different environments.

The application wrapper can request input files from the controller to be staged in the execution directory before the application is run/re-run. This is specified using the following arguments:

`-in` - Requests input files from the controller prior

to application execution.

`-append -` Same as `-in` except incremental updates to the input files are repeatedly requested during application execution.

As with the equivalent arguments output files (see Section 4.1.1), the above commands take a file list detailing the files to be requested. This file list can include the keyword `STDIN` to indicate requesting the standard input.

4.2 Application Controller

A gridMonSteer application controller is a process that receives input requests and output notification from the application wrapper via a network interface. An application controller is generally run locally to the user, allowing the output of an application to be visualized on the users desktop for example, however it is equally valid for a remote application controller to be used. Also, an application controller can be used to control multiple jobs. These jobs can be independent applications or separate parts of a parallel application.

In the current implementation application controllers are exposed as web services, however the gridMonSteer architecture could function equally effectively using an alternative communication framework. In this implementation, the controller is required to expose an interface containing six operations (*runJob, getInput, notifyOutput, notifyState, registerOutput, selectOutput*), and providing this interface is exposed, the actual behavior of the controller is undefined.

In the scenarios we are currently developing (see Section 5 for an example), we use WSPeer [18] to allow the application controller to dynamically expose itself as a web service for the duration of the jobs it is managing. Although this is beneficial for developing interactive application controllers, any web service implementing the gridMonSteer controller interface could be employed as an application controller.

5 Case Study: Using Cactus to Steer Triana Workflows

In this case study we illustrate how gridMonSteer can be used to steer complex Grid workflows. Here, we consider the case that the workflow itself is the Grid application rather than the current perception that the legacy code being invoked remotely is the application. Although, this example is simple, it could be extended to data driven applications that involve complex decision-based workflow interactions across the resources.

The specific scenario we illustrate in this case study is a gridMonSteer wrapped Cactus [10][19] job executed within a Triana workflow [8][9], with Triana both submitting the Cactus job to the Grid resource manager and acting as gridMonSteer controller for that job. As controller, Triana receives timely notification of intermediate results via gridMonSteer that are then used to steer the overall workflow being executed.

The original idea for gridMonSteer came from a SC04 demonstration in which Triana was used to visualize two orbiting sources simulated by a Cactus job running in a Grid environment [20]. Accessing intermediate results is a typical requirement for Cactus users who want to monitor the progress of their application or derive scientific results. For this scenario a specific Cactus thorn was coded to notify the files created by Cactus to the Triana controller after each time step, however this is a Cactus specific solution. In this example we recreate this demonstration except Cactus is treated as a true legacy application, and gridMonSteer provides non-intrusive monitoring and steering capabilities.

Cactus simulations present a number of interesting cases for gridMonSteer. The output from Cactus can be in a number of formats, including HDF5, JPEG, and ASCII formats suitable for visualizing with tools such as X-Graph and GNUPlot. In some instances, such as JPEG images, the output files are overwritten after each time step, where other output files are appended, such as X-Graph files.

Grid job submission in Triana is conducted via the GridLab GAT [5], a high level API for accessing Grid services. The GAT uses a pluggable adaptor architecture that allows different services to be used without modifying the application code. Current resource manager adaptors for the Java GAT include GRAM, GRMS and local execution. Although different adaptors can be used within the GAT, as gridMonSteer is generic to any resource manager, Triana can run gridMonSteer wrapped jobs via the GAT without having to modify the job submission to take account of different resource

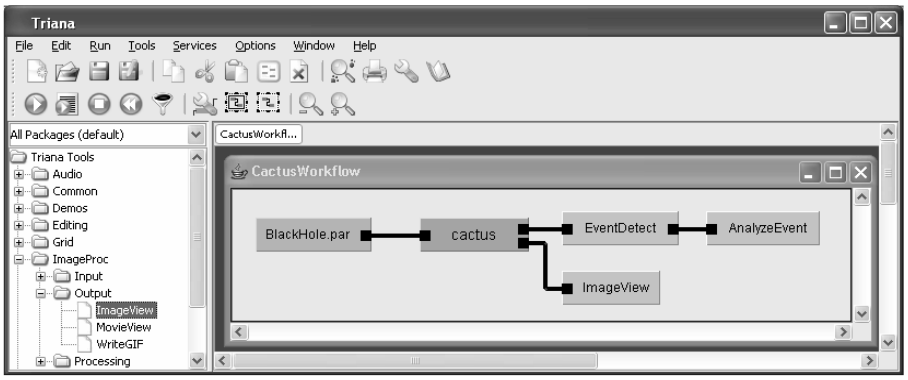


Figure 2: An example Triana workflow for executing a gridMonSteer wrapped Cactus job.

manager bindings.

When submitting a Cactus job, the arguments specified for the gridMonSteer application wrapper by Triana include:

`-xmonitor:*.jpg` - Monitor for creation and modification of JPEG files.

`-xupdate:*.xg;STDOUT` - Monitor for updates to the X-Graph ASCII files and standard output.

`-state` - Notify the state of the Cactus job (started, stopped etc.).

These arguments cover the required monitoring and steering for a Cactus job, as described above.

In Figure 5 we show a gridMonSteer wrapped Cactus job embedded in an example Triana workflow. Data flow within Triana is represented left to right, so in this example BlackHole.par is providing the input to the Cactus job. As BlackHole.par is a File object, this represents a pre-staged file for the Cactus job.

As the select/register arguments (`-xmonitor`, `-xupdate`) are specified for the gridMonSteer wrapper, the outputs from the Cactus job are registered with Triana when they are created/updated. A simple user interface allows the scientist to interactively select from the available outputs and map them to output nodes on the Cactus workflow task. This can be used for example to change the simulation elements they are studying if they observe an interesting event. The selected outputs are immediately notified to Triana by the application wrapper and used to drive the remaining workflow.

In the example shown in Figure 5, there are two outputs from the Cactus workflow task. One output is directed to ImageView, a local component that enables the visualization of the image files generated by the running Cactus job. The other output is directed to EventDetect, which represents the tools we are developing within Triana to detect interesting events in Cactus output data, such as the apparent horizon of a black hole simulation. When an interesting event is detected it can be analyzed locally (as represented by AnalyzeEvent), or alternatively further Grid jobs can be launched to process the event remotely.

6 Conclusion

In this paper we have outlined gridMonSteer, a generic architecture for monitoring and steering legacy applications in Grid environments. Unlike existing Grid monitoring solutions, the gridMonSteer approach is non-intrusive and therefore does not require any modification to the application code. Furthermore, as the gridMonSteer application wrapper is executed as a standard job, it is generic to any Grid resource manager and can be used within a Grid service framework. All communication is initiated by the application wrapper and is therefore outbound from the Grid resource, a situation that is generally allowed by firewalls as it implies the job has already complied with the security requirements of that resource. The behavior of the application controller that is employed to handle information requests/notification from the

wrapper can be tailored to individual or generic application requirements.

In this paper we presented a case study using gridMonSteer to monitor a legacy application running as a component within a Grid workflow. In this case study, the intermediate results from the legacy application where notified by gridMonSteer to the workflow controller and used to dynamically steer the workflow execution. This is an example of how gridMonSteer enables interactive legacy applications to be used within Grid environments in situations that were previously unavailable.

References

[1] S. Zanikolas and R. Sakellariou, "A taxonomy of grid monitoring systems," *Future Generation Computer Systems*, vol. 21, pp. 163–168, January 2005.

[2] Z. Balaton and G. Gombas, "Resource and job monitoring in the grid," in *Proceedings of the Ninth International Euro-Par Conference, Vol. 2790 of Lecture Notes in Computer Science*, Springer-Verlag, 2003.

[3] B. Balis, M. Bubak, T. Szepieniec, R. Wismuller, and M. Radecki, "Monitoring grid applications with grid-enabled OMIS monitor," in *Proceedings of the First European Across Grids Conference, Vol. 2970 of Lecture Notes in Computer Science*, pp. 230–239, Springer-Verlag, 1997.

[4] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke, "A Resource Management Architecture for Metacomputing Systems," in *Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing*, pp. 62–82, IEEE Computer Society, 1998.

[5] "The GridLab Project." See web site at: `http://www.gridlab.org`.

[6] J. Frey, T. Tannenbaum, M. Livny, I. Foster, and S. Tuecke, "Condor-G: A Computation Management Agent for Multi-Institutional Grids," in *Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPCD-'01)*, 2001.

[7] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," tech. rep., Open Grid Service Infrastructure WG, Global Grid Forum, 2002.

[8] I. Taylor, M. Shields, I. Wang, and O. Rana, "Triana Applications within Grid Computing and Peer to Peer Environments," *Journal of Grid Computing*, vol. 1, no. 2, pp. 199–217, 2003.

[9] "The Triana Project." See web site at: `http://www.trianacode.org`.

[10] T. Goodale, G. Allen, G. Lanfermann, J. Masso, T. Radke, E. Seidel, and J. Shalf, "The cactus framework and toolkit: Design and applications," in *Vector and Parallel Processing VECPAR 2002, 5th International Conference, Lecture Notes in Computer Science*, Springer, 2003.

[11] "The Cactus Computational Toolkit." See web site at: `http://www.cactuscode.org`.

[12] D. M. Beazley and P. S. Lomdahl, "Lightweight Computational Steering of Very Large Scale Molecular Dynamics Simulations," in *Proceedings of Super Computing '96*, 1996.

[13] Y. Huang and D. W. Walker, "JACAW - A Java-C Automatic Wrapper Tool and its Benchmark," in *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)*, 2002.

[14] T. Delaitre, A.Goyeneche, P. Kacsuk, T.Kiss, G. Terstyanszky, and S. Winter, "GEMLCA: Grid execution management for legacy code architecture design," in *Proceedings of the 30th EUROMICRO Conference, Special Session on Advances in Web Computing*, August 2004.

[15] M. Senger, P. Rice, and T. Oinn, "A resource management architecture for metacomputing systems," in *Proceedings of UK e-Science All Hands Meeting*, pp. 509–513, September 2003.

[16] R. Buyya, D. Abramson, and J. Giddy, "Nimrod/g: An architecture of a resource management and scheduling system in a global computational grid," in *HPC Asia 2000*, pp. 283–289, May 2000.

[17] "Nimrod: Tools for distributed parametric modelling." See web site at: `http://www.csse.monash.edu.au/ ?davida/nimrod/`.

[18] A. Harrison and I. Taylor, "WSPeer - An Interface to Web Service Hosting and Invocation," in *HIPS Joint Workshop on High-Performance Grid Computing and High-Level Parallel Programming Models*, 2005.

[19] "The Cactus Project." See web site at: `http://www.cactuscode.org`.

[20] T. Goodale, I. Taylor, and I. Wang, "Integrating Cactus Simulations within Triana Workflows," in *Proceedings of 13th Annual Mardi Gras Conference - Frontiers of Grid Applications and Technologies*, pp. 47–53, Louisiana State University, February 2005.

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

SolarB Global DataGrid

Tim Folkes[1], Elizabeth Auden[2], Paul Lamb[2], Matthew Whillock[2], Jens Jensen[1],Matthew Wild[1]

1 CCLRC – Rutherford Appleton Laboratory (RAL)
2 Mullard Space Science Laboratory (MSSL), University College London

Abstract

This paper describes how telemetry and science data from the Solar-B satellite will be imported into the UK AstroGrid infrastructure for analysis. Data will be received from the satellite by ISAS in Japan, and will then be transferred by MSSL to the UK where it is cached in CCLRC's Atlas tapestore, utilizing existing high speed network links between RAL and Japan. The control of the data flow can be done from MSSL using their existing network infrastructure and without the need for any extra local data storage. From the cache in the UK, data can then be redistributed into AstroGrid's MySpace for analysis, using the metadata held at MSSL. The innovation in this work lies in tying together Grid technologies from different Grids with existing storage and networking technologies to provide an end-to-end solution for UK participants in an international scientific collaboration. Solar-B is due for launch late 2006, but meanwhile the system is being tested with data from another solar observatory.

Introduction

Solar-B is the latest Japanese solar physics satellite and is due for launch in late 2006. Its instrumentation includes a 0.5m Solar Optical Telescope (SOT), an Extreme Ultraviolet (EUV) imaging spectrometer (EIS) [EIS] and an X-ray/EUV telescope (XRT). The instruments will work together as a single observatory. The SOT has an angular resolution of 0.25" and covers a wavelength range of 480-650nm; it includes the Focal Plane Package (FPP) vector magnetograph and spectrograph. The vector magnetograph will provide time series of photospheric vector magnetograms, Doppler velocity and photospheric intensity. XRT will provide coronal images at different temperatures; both full disk and partial fields of view. EIS will provide monochromatic images of the transition region and corona at high cadence using a slot. High spectral resolution images can be obtained by rastering with a slit.

Telemetry from the SolarB satellite will be captured in Japan then relayed via gridftp third-party transfer to Rutherford Appleton Laboratory (RAL). Data pipeline processing, cataloguing and user data access will be carried out at MSSL but all project data other than catalogues will be held solely at RAL.

The data will be stored in the Hierarchical Storage Management (HSM) system at RAL. This consists of an SGI Altix 350 system running the Data Migration Facility (DMF) software [DMF]. This software manages the movement of the data to a hierarchy of disk and tape. It also manages the space in the filesystem.

A user interface has been developed that is based on the Astrogrid software being run by the UK Solar System Data Centre (UKSSDC) at RAL. Again, the information about the files is held at MSSL, and the data will move direct to the user from RAL.

Data flow

The diagram shows the interaction between users in the solar research community, the instrument and the ground segment. Proposals are submitted for observations ("studies") which, once accepted, are

integrated by the Japanese Institute for Space and Astronautical Science (ISAS) into the satellite observation programme. Once data from the Extreme Ultraviolet Spectrometer (EIS) is received by the ground segment at ISAS it is reformatted as raw ("level 0") FITS files then transferred to the Atlas DMF at RAL using GridFTP [GridFTP]. Data from other instruments will be similarly transferred, as level 0 and level 2 FITS files. Following transfer, a notification is sent to MSSL containing details of the files transmitted. The Catalogue Manager at MSSL then accesses the DMF to retrieve recent EIS level 0 data.

The EIS data processing pipeline will run at MSSL. Here, level 0 data is processed into level 1 (intermediate) and level 2 (science products). The resulting FITS files are sent to the DMF for storage, again using GridFTP.

The MSSL Catalogue Manager will periodically poll the DMF to ensure that the DMF does hold two archive copies of all data Solar-B data. Once this is confirmed for the EIS level 1 and level 2 files, they can be deleted from the MSSL cache. More generally, it is necessary for the Catalogue Manager to hold sufficient status information about Solar-B data within the DMF so as to be able to process user data requests.

MSSL will act as the UK point of contact for research community access to all Solar-B science data. The community will use a web-based system to access the catalogue and generate commands to transmit science products from the DMF into the user's MySpace. This is described in the next section.

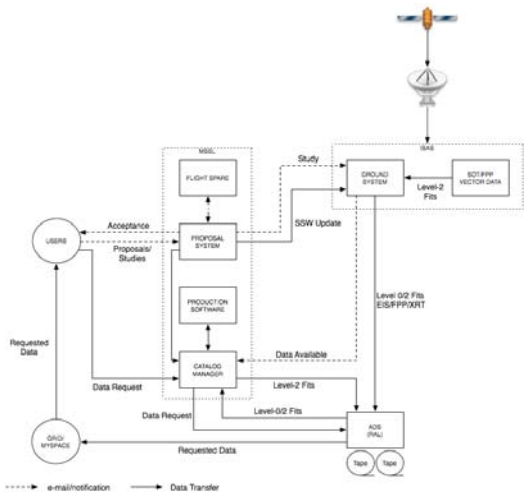


Figure 1 Solar-B data flow

Solar-B User Access

The Solar-B user access prototype exhibits functionality that will allow UK solar users to execute an AstroGrid [AstroGrid] workflow that performs two tasks. First, a data search is executed to identify Solar-B stored in the ATLAS datastore facility. Second, files matching the user's request are transferred to the user's AstroGrid MySpace area via a third party transfer. For the prototype, EIT files from SOHO (Solar and Heliospheric Observatory, [SOHO]) have been uploaded to the ATLAS datastore. The FITS header keywords from these files have been transferred to a MySQL database hosted at MSSL. The production version of the user interface will access the files direct from the NFS mounted DMF controlled disk.

To enable data searching, an AstroGrid DataSet Access (DSA) module has been installed on a web-visible machine at MSSL and configured for access to the prototype database. The DSA module and database exist on separate machines to represent a publicly visible DSA instance integrated with a database on a machine that is not publicly visible; however, these two pieces of software can be deployed on the same machine if desired. Once the DSA module can successfully communicate with the database, the DSA is registered with an AstroGrid registry; registration makes the

Solar-B dataset searchable by the virtual observatory. Users can query the Solar-B metadata by submitting an Astronomical Data Query Language (ADQL) [ADQL] query, an XML packet that is translated to SQL by the DSA module. The query's result is an XML file in VOTable format [VOTable] containing metadata from the MSSL Solar-B database, such as FITS (astronomical and image metadata) header keywords and file location as a URL.

Data searches can be executed as a step in an AstroGrid workflow, and data retrieval is a logical second step. Although the Solar-B files held in ATLAS are not publicly visible, a publicly visible machine within the UKSSDC can stage files in a web accessible area. This machine hosts a servlet that, when called, executes the ATLAS tape command with tapeID, file number, file name and file owner as input parameters. In the production version the machine can mount the Solar-B DMF area as a read-only filesystem and generate the necessary URLs to access the data. The file is retrieved from ATLAS to the staging machine and is visible to the user as a URL. The user can transfer many files to the AstroGrid MySpace by running an AstroGrid Client Runtime (ACR) script from the AstroGrid workbench that uses the `astrogrid.ioHelper.getExternalValue` method to parse through a list of URLs supplied from the VOTable in the data search described above. The `getExternalValue` method creates an input stream from the contents of each URL that spawns the retrieval of files from ATLAS. An output stream is created for a new file in MySpace, and the contents of the input stream are written to the output stream. The user can then browse the retrieved Solar-B files in his or her MySpace browser.

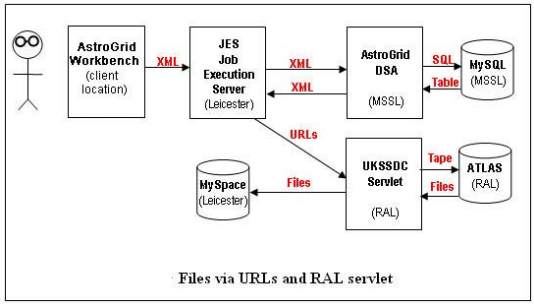


Figure 2 User access

Security

The data that is ingested into the archive comes from sites in Japan and the US. Tapestores usually support two groups of data transfer protocols: a local and a remote one. The local protocol is relatively lightweight with a simple authentication mechanism, and no data encryption. Remote protocols often provide encryption, and almost always have secure authentication methods.

Generally, for security reasons it is not appropriate to use the local protocol over the wide area network (WAN). It has often been done in the past, mainly for convenience, but sites are getting increasingly worried about protecting themselves and their tapestore facilities, and users are getting worried about access to and protection of their data. Even users who do not require encrypted transfers (data confidentiality), still need to know that the data has been written by an authenticated user, and that the data has not been altered in transit either by mistake or by a malicious attacker.

The Atlas datastore local protocol does not provide sufficient security for this, so in this project we decided to use GridFTP [GridFTP] to transfer data across the WAN. GridFTP combines Globus Grid security [GSI] with good file transfer performance. In the security model both users and hosts authenticate to each other using X.509 certificates [X509].

Another advantage of GridFTP is the ability to do "third party" transfers. This means that users at MSSL can transfer data

from the Japanese site, ISAS, to RAL, without having to copy the data to MSSL over slower network links.

To set up this transfer, we needed to identify appropriately trusted Certification Authorities in the relevant countries, to issue certificates to all the users and hosts. This was easy for the UK and Japan. Both countries have national Grid CAs: the UK e-Science CA [UKCA] and its Japanese counterpart [AIST]. Since CAs know and trust each other, it was easy to get the CAs to establish trust on behalf of the users. The US is slightly more tricky because there is no CA covering all of the US, but the Japan-UK links was highest priority. Lockheed, one of the US partners, will be transferring data initially to ISAS, and from there it will be replicated to the UK via the GridFTP link.

Using the existing Grid CAs is by far the best option because the CAs have already negotiated trust globally [IGTF].

RAL data storage

For this project the data will be stored in a Hierarchical Storage Management (HSM) system. The HSM uses the Data Migration Facility (DMF) [DMF] software provided by Silicon Graphics Inc. (SGI). The hardware is based on a 4 processor Atltix 350 system running Suse linux and the SGI ProPack software.

The DMF system allows the users or as in this case the application to just use the filesystem provided without having to worry about managing the space or worrying about the security of the data.

The DMF software fulfils two roles. The first is the migration of data from the primary disks to (in our case) other hierarchies of disk or tape. The second function is the automatic management of the space in the file-systems.

To migrate the data to other storage hierarchies, policies are created that use various Media Specific Processes (MSP) to move the data. At RAL we have configured the system to create two copies of all the data. If the file is smaller that 50MB then the first copy of the data is written onto another set of disk. This allows for fast retrieval of small files. The second copy of the data goes onto tape. For all other file sizes, the data is written to two separate sets of tape. In both cases the second set of tapes are removed from the robot and are stored in a fire-safe.

Once the files have been migrated to a MSP, the file becomes "dual state" with a copy of the data on disk and at least one else-where. This is the first stage in the data migration process.

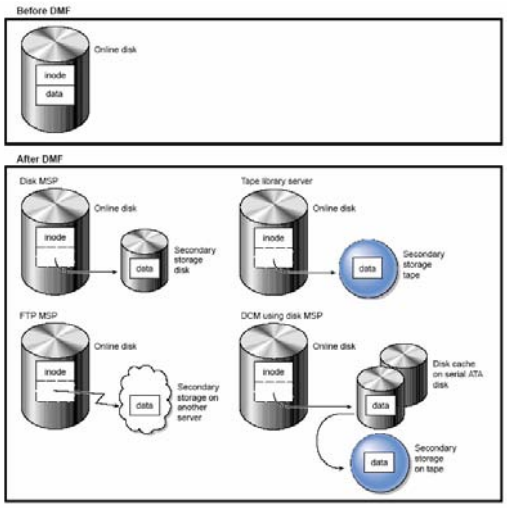


Figure 3 DMF. Copyright SGI Inc 2006 sgi

To manage the free space in the file-systems a series of policies are created that specify the levels at which the percentage space is to be maintained in the filesystem. These levels are "free space minimum", "free space target" and "migration target".

- "free space minimum" specifies the minimum percentage of filesystem space that must be free. When this value is reached, DMF will take action to migrate and free enough files to bring the filesystem into line with the policy.

- "free space target" specifies the percentage of free filesystem space DMF will try to achieve if free space falls below "free space minimum".
- "migration target" specifies the percentage of filesystem capacity that is maintained as a reserve of space that is free or occupied by dual-state files. DMF attempts to maintain this reserve in the event that the filesystem free space reaches or falls below "free space minimum".

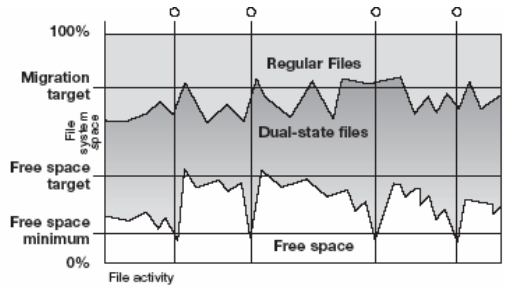


Figure 4. Filesystem space management.

This is the second stage of the data migration process. Once a file is dual state it is easy for DMF to free up the space, as it already has a copy of the file on an MSP, so the data can be deleted from disk.

The recall of the files is again automatic. As soon as there is an open request for the file, the kernel will direct the request to the DMF system. The open will wait until the system has recalled the data from the MSP.

The files look like normal files either locally or via NFS. If a user wants to see what state a file is in from the DMF point of view, then DMF provides the "dmls" or "dmstat" commands. These are available as client commands on some versions of Linux.

The file-systems are backed up using a modified dump programme that is aware of the state of the files. If a file is dual state i.e. it has a copy on disk and a copy in a MSP (tape for example) then the dump program will only backup the inode information. If the filesystem, or individual file, is restored from the backup, then the file will be in a off-line state, and when the user access the file the data will be restored from the MSP. Care has to be taken to make sure the backups and the backups of the DMF database are kept in step.

DMF also allows files that are accidentally deleted to be recovered. When a file is deleted, it is removed from disk, but an entry is still kept in the DMF database. The files become "soft deleted". After a pre-determined period (30 days by default) the soft-deleted entries are hard deleted and the space in the MSP freed up. After this there is no way to recover the file.

Conclusion

In this paper we have demonstrated how we have built an international datagrid for UK participation in the SolarB project. Using existing data management components, data is transferred securely from the downlink in Japan to the UK where it is cached at RAL's Atlas Petabyte Store. From there it is then imported into AstroGrid's MySpace for analysis.

The whole process is controlled from MSSL at UCL.

References

[ADQL] Ohishi, M. and Szalay, A. "IVOA Astronomical Data Query Language." Version 1.01. 24 June 2005. http://www.ivoa.net/Documents/latest/ADQL.html

[AIST]https://www.apgrid.org/CA/AIST/Production/index.html

[AstroGrid] Rixon, G. "Software Architecture of AstroGrid v1", 27 September 2005. http://software.astrogrid.org/developerdocs/astrogrid-v1.1-architecture.htm

[DMF] http://www.sgi.com/products/storage/tech/dmf.html

[EIS] "The Solar-B EUV Imaging Spectrometer and its Science Goals", J.L. Culhane, L.K. Harra, G.A. Doschek, J.T. Mariska, T. Watanabe, H. Hara, 2005, Adv. in Space Res, Vol 36,1494 - 1502.

[GSI] http://www.globus.org/toolkit/security

[IGTF] http://www.gridpma.org/

[SGI] http://www.sgi.com

[SOHO] http://sohowww.nascom.nasa.gov/

[UKCA] http://www.grid-support.ac.uk/ca/

[VOTable] Ochsenbein, F. et al. "VOTable Format Definition." Version 1.1. 11 August 2004. http://www.ivoa.net/Documents/latest/VOT.html

[X509] http://www.grid-support.ac.uk/content/view/83/42/

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

eSDO Algorithms, Data Centre and Visualization Tools for the UK Virtual Observatory

Elizabeth Auden[1], J.L. Culhane[1] Y.P. Elsworth[2], A. Fludra[3], M.J. Thompson[4]

[1]Mullard Space Science Laboratory, University College London.
[2]Physics Department, University of Birmingham.
[3]Rutherford Appleton Laboratory
[4]Department of Applied Mathematics, University of Sheffield

Abstract

The eSDO project is funded by PPARC to make data, algorithms and visualization techniques from the Solar Dynamics Observatory mission available to the UK solar community through the virtual observatory. Scientists and developers based at MSSL, RAL and the Universities of Sheffield and Birmingham form a consortium that has spent one year assessing the UK solar community's requirements for SDO access. Developers are now engaged in a two year implementation phase that will deliver three workpackages that encompass ten solar algorithms, UK data centre designs and visualization tools in preparation for SDO launch in August 2008.

1. Introduction

The Solar Dynamics Observatory (SDO) mission will be launched on 31 August 2008 with a payload of three instruments: the Atmospheric Imaging Assembly (AIA), the Helioseismic and Magnetic Imager (HMI) and the EUV Variability Experiment (EVE) [1]. The data rate from these instruments will approach 2.5 TB / day; this volume of data presents new challenges to the global solar community. Although the primary SDO data centre will be maintained in the US by the Joint Science Operations Center (JSOC), a secondary UK data centre will allow UK solar scientists to search for and retrieve data through the AstroGrid virtual observatory (VO) [2]. Visualization tools such as catalogues, streaming image viewers and movie makers will allow scientists to identify pertinent datasets more rapidly; once selected, these datasets can be processed with VO-enabled solar algorithms.

The ten algorithms under development by the eSDO project fall into two categories: event / feature recognition and helioseismology. The first category of algorithms will be primarily used by event driven solar physicists studying flares, coronal mass ejections (CMEs) and other solar behaviours displaying rapid evolution. The second category of algorithms, helioseismology, will be used by scientists examining the interior of the Sun.

The needs of these two user groups direct the design of the UK SDO data centre. Event driven scientists expect access to data as soon as an interesting solar event has occurred, so rapid searching and avoidance of network bottlenecks are priorities for this user group. In contrast, helioseismologists study solar oscillations over much longer periods of time. While they do not require immediate access to data following solar events, scientists in the helioseismology community typically process months or years of solar data at a time. This group's priorities are storage and processing capabilities for large blocks of data.

2. Algorithms

2.1 Solar Algorithms

The eSDO project will deliver C code and documentation for four helioseismology algorithms and six feature / event recognition algorithms. Developers at MSSL and RAL are concentrating on the first category of algorithms, which will primarily process data

from the AIA instrument. These algorithms include non-linear magnetic field extrapolation, coronal loop recognition, helicity computation, small event detection and recognition of coronal mass ejection (CME) dimming regions. The magnetic field extrapolation algorithm will particularly benefit from grid computing techniques. Wiegelmann's "optimization" method of 3-D magnetic field extrapolation is used to calculate magnetic field strength over a user-defined solar volume [3]. Computations of volume integrals are distributed with the MPICH 1 or 2 parallel processing protocol; current investigations are exploring the execution of this code on National Grid Service (NGS) facilities through an AstroGrid application interface.

The universities of Birmingham and Sheffield are examining global and local helioseismology respectively, and these algorithms will process data from the HMI instrument. The Birmingham group is implementing mode frequency analysis and mode asymmetry analysis algorithms. The Sheffield group is developing subsurface flow analysis, perturbation map generation and computation of local helioseismology inversion algorithms. Helioseismology algorithms analyse low frequency solar oscillations with a periodicities that range from 5 minutes for *p* mode (acoustic) waves to hours or days for *g* mode (gravity) waves [4]. These oscillations are studied over long periods of time, so helioseismology algorithms require correspondingly long data sequences as input. Because of the large input series, these algorithms should be hosted to locally to helioseismology data caches in the US or UK to ensure efficient completion of data processing.

2.2 Algorithm Distribution

Solar algorithms developed by eSDO will be made available to UK users through the JSOC pipeline, AstroGrid, and SolarSoft. The JSOC pipeline systems at Stanford University and Lockheed Martin will contain processing modules that either run automatically or when invoked by user requests. Most data products created through the automated modules will be made available to the public through the virtual observatory; however, authorized pipeline users can execute optional modules if higher level processing or an alternative implementation of an algorithm is required. Some eSDO code, such as the global helioseismology mode parameters algorithm, will be integrated with the JSOC pipeline as automated modules. Other eSDO algorithms will be designated by JSOC as optional user-invoked modules. UK SDO co-investigators and their teams will be able to access the JSOC pipeline directly through accounts at Stanford University. The coronal loop recognition algorithm will be the eSDO test case for JSOC pipeline integration.

For the wider UK solar community, access to eSDO algorithms will be available through AstroGrid or SolarSoft. Algorithms deployed as UK hosted AstroGridT CEA web services will be accessible through the AstroGrid workbench via the application launcher and parameterized workflows. Emerging VO access to facilities such as NGS will allow scientists to execute computationally intensive algorithms on powerful remote machines. Algorithms that are not computationally intensive will be wrapped in IDL for SolarSoft distribution through the MSSL gateway. CEA deployment will begin in late September 2006.

In addition to the algorithms under development by eSDO consortium institutions, the project will also work with UK coronal seismologists to deploy wave power analysis algorithms as JSOC pipeline modules and AstroGrid CEA applications.

3. Data Visualization

The huge volume of SDO data collected every day makes it imperative to search the data archive efficiently. The visualization techniques including streaming tools, catalogues, thumbnail extraction and movie generation will aid scientists in archive navigation. By enabling scientists to identify relevant datasets quickly, these tools will reduce network traffic and save research time.

3.1 Browse Products

Browse products are files such as thumbnail images and movies that scientists can view quickly to identify interesting datasets that merit further processing. Thumbnail images of selected AIA and HMI datasets will be available through VSO, and the JSOC will produce movies of "regions of interest". The browse product tools developed by eSDO will allow scientists to generate their own thumbnail image galleries and movies through a web interface. Users will specify start time, end time, cadence, and data products along with outputs of either an image gallery or a movie. A web service will

locate the datasets, extract images from FITS files, label the images with relevant metadata from the FITS headers, and then display an image gallery in the user's web browser or create an MPEG movie that the user can view or download. Following advice from the eSDO Phase A review with PPARC in November 2005, the movie maker will be enhanced to evaluate a data sequence's suitability for wavelet analysis. Prototypes of the thumbnail maker and image gallery tool are now available [5,6].

3.2 Catalogues

Two science catalogues will be generated from eSDO algorithms. One catalogue will store statistical information about small solar events and CME dimming regions. The catalogue will be generated in the UK and annotated continuously from the UK data centre's cached AIA datasets. The second catalogue will provide a "GONG month" [7] of helioseismology information produced by the mode parameters analysis algorithm. This catalogue will be generated in the US using eSDO modules integrated with the JSOC pipeline. Both catalogues will be searchable through the virtual observatory with instances of the AstroGrid DataSet Access (DSA) software.

3.3 SDO Streaming Tool

The SDO streaming tool will display HMI and AIA science products in a web GUI. Scientists will be able to pan and zoom the images in both space and time. A user will launch the java webstart SDO streaming tool from a web browser and then specify a start time, stop time, cadence, and data product. Three types of SDO products will be available: AIA images from 10 channels, HMI continuum maps, and HMI line-of-sight magnetograms.

The user will be able to zoom in spatially from a full disk, low resolution image to a full resolution 512 by 512 pixel display of a solar area. Users will also be able to pan spatially zoomed images in eight directions. Once a user has selected a cadence (for instance, 1 image per hour), data products matching that cadence will be displayed on the screen; users will be able to "zoom" in time by increasing or decreasing this cadence, while rewind, fast forward and pause facilities will allow users to "pan" in time.

Collaboration with the UK coronal seismology community will enhance the streaming tool with a wave power analysis function.

A prototype of the eSDO streaming tool that can be used in conjunction with the AstroGrid Helioscope application is available now [8].

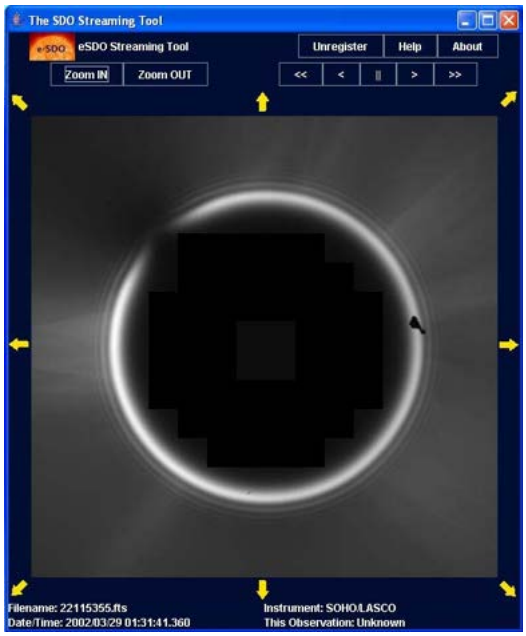


Figure 1: The SDO streaming tool prototype loading a coronal image from the SOHO LASCO instrument, zoomed in one spatial level.

4. UK Data Centre

The primary SDO data centre will be based in the US at Stanford University and Lockheed Martin, and users will be able to access the data stored there directly through the Virtual Solar Observatory (VSO). A second data centre will be established in the UK at the ATLAS datastore facility at the Rutherford Appleton Laboratory. Rather than providing a full data mirror, this data centre will store all SDO metadata as well as holding a cache recent data products and popular data requests. The metadata storage will allow SDO data searches to be included in AstroGrid workflows while the local data cache will provide UK scientists with access to SDO data when the US data centre experiences decreases in network speed during periods of high user demand.

4.1 Metadata and VO Searches

JSOC will store SDO metadata in a distributed Data Resource Management System (DRMS) while actual data files will be stored in a distributed Storage Unit Management System (SUMS). Each instance of DRMS will hold metadata for all SDO files, regardless of the fraction of SDO data stored in the local SUMS. When new SDO high level data sets are created in the UK, the metadata for those datasets will be entered into a DRMS database and disseminated globally to all DRMS instances. Instances of DRMS operate in conjunction with Postgres and Oracle databases to store keyword value pairs that are used to generate SDO FITS headers on the fly.

JSOC has designated the Virtual Solar Observatory (VSO) as the official front end for user access to SDO data, and UK scientists will be able to download SDO data from the VSO web interface. However, VSO is not compliant with the International Virtual Observatory Alliance (IVOA) web services that AstroGrid uses; therefore AstroGrid workflows cannot access SDO data through VSO. Instead, the eSDO project is configuring an instance of the AstroGrid DSA software to interface with the UK DRMS. This will allow scientists to set up an AstroGrid workflow that will search through the SDO metadata, identify the relevant SDO datasets, and retrieve the datasets as URLs.

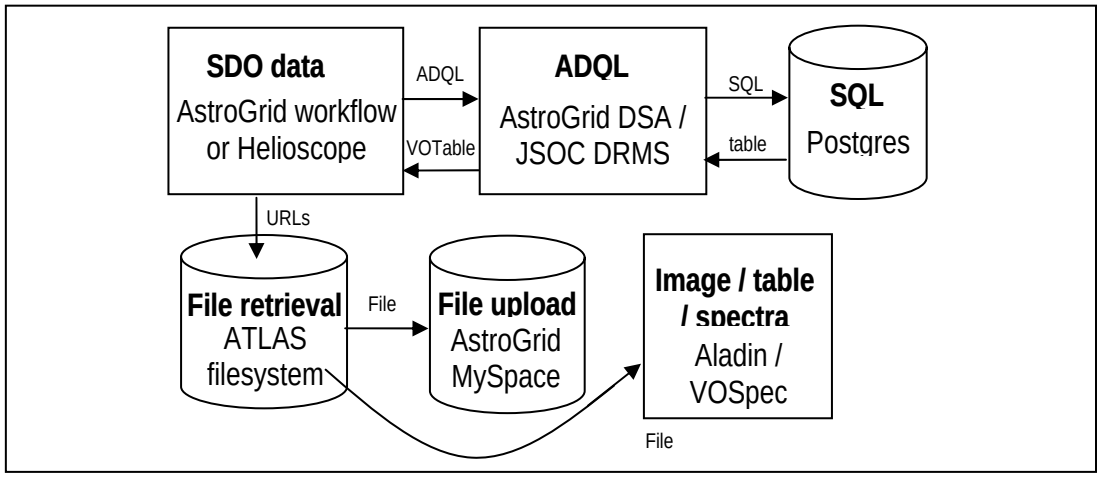


Figure 2: This figure demonstrates the flow of information when an AstroGrid user submits a request for AIA or HMI data to the UK SDO data centre.

4.2 ATLAS Data Cache

Although a UK instance of the DRMS server will provide a UK holding of all SDO metadata, the datasets themselves will be archived in the US. The eSDO project has proposed a 30 TB disk cache hosted at the ATLAS facility for holding a fraction of the SDO data.15 TB will hold a rolling 60 day cache of recent AIA and HMI data while the other 15 TB will serve as a "true cache" for older datasets requested by UK users through the AstroGrid system. When a UK user requests an HMI or AIA science product through the AstroGrid system, the request will be sent to the UK data centre. If the data is not available there, the request will be redirected through an AstroGrid / VSO interface, and the relevant data products will be returned to the UK in export format (FITS, JPEG, VOTable, etc).

In the event of a large flare or coronal mass ejection, UK scientists can download data from the rolling 60 day cache without impedance from the heavy network traffic of users accessing the JSOC archive. In addition, UK users of the SDO streaming tool will have more seamless access to images viewed at a high cadence. The true cache will reduce transatlantic data transfers for such user groups as the helioseismology community that process large blocks of data but are not concerned with recent events.

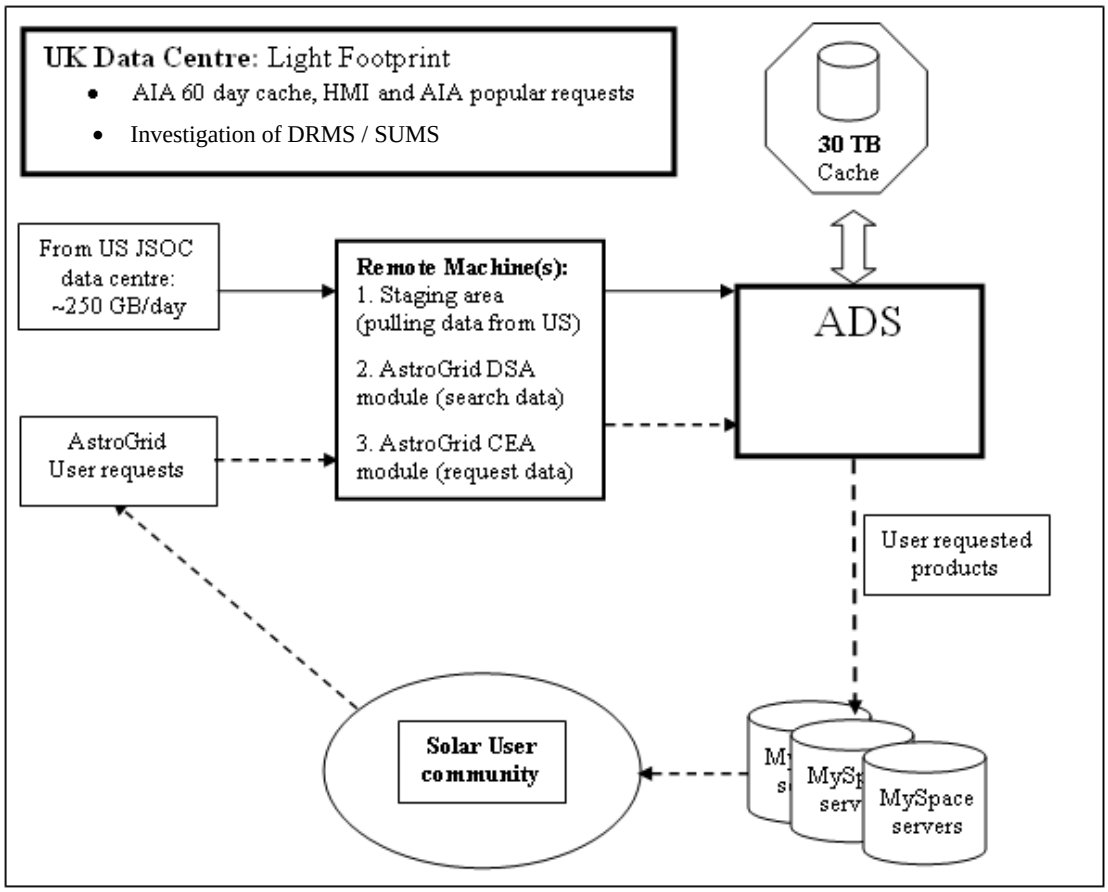


Figure 3: The proposed architecture for a UK SDO data centre demonstrates the interaction between the full US SDO archive, the UK cache, and user requests.

5. Conclusions

Although eSDO funding will cease before the SDO mission is launched in August 2008, the project will deliver completed designs, code and documentation for each of the three major workpackages. Ten algorithms concentrating on solar feature recognition and helioseismology will be ready for deployment through the JSOC pipeline, AstroGrid and SolarSoft. The UK data centre will provide interfaces between the JSOC data resource management system, the AstroGrid data set access software and the ATLAS storage facility, allowing execution of virtual observatory searches all SDO metadata with rapid access to SDO data cached in the UK. Finally, scientists will be able to identify relevant datasets through visualization tools such as catalogues, web-based image gallery and movie makers, and the SDO streaming tool.

6. References

[1] "Solar Dynamics Observatory Mission Factsheet", 3 November 2005. http://sdo.gsfc.nasa.gov/sdo_factsheet.pdf

[2] Rixon, G. "Software Architecture of AstroGrid v1", 27 September 2005.

[3] Wiegelmann, T. 2004, Solar Physics, 219, 87-108

[4] Lou, Y.Q. 2001, Astrophysical Journal, 556, 2, L1221-L125

[5] Auden, E. eSDO Thumbnail Maker, 19 July 2006. http://msslxx.mssl.ucl.ac.uk:8080/esdo/visualization/jnlp/esdo_thumbnailmaker.jnlp

[6] Auden, E. eSDO Image Gallery, 18 July 2006. http://msslxx.mssl.ucl.ac.uk:8080/esdo/visualization/jnlp/esdo_imagegallery.jnlp

[7] Howe, R., Komm, R., Hill, F. 2000, Solar Physics, 192, 2, 427-435

[8] Auden, E. eSDO Streaming Tool, 14 July 2006. http://msslxx.mssl.ucl.ac.uk:8080/esdo/visualization/jnlp/esdo_streamingtool.jnlp

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

e-Science Tools For The Genomic Scale Characterisation Of Bacterial Secreted Proteins

Tracy Craddock[1], Phillip Lord[1], Colin Harwood[2] and Anil Wipat[1]

[1]School of Computing Science and [2]Cell and Molecular Biosciences, Newcastle University, Newcastle upon Tyne, Tyne and Wear, UK

Abstract

Within bioinformatics, a considerable amount of time is spent dealing with three problems; *heterogeneity*, *distribution* and *autonomy* – concerns that are mirrored in this study and which e-Science technologies should help to address. We describe the design and architecture of a system which makes predictions about the secreted proteins of bacteria, describing both the strengths and some weaknesses of current e-Science technology.

The focus of this study was on the development and application of e-Science workflows and a service oriented approach to the genomic scale detection and characterisation of secreted proteins from *Bacillus* species. Members of the genus *Bacillus* show a diverse range of properties, from the non-pathogenic *B. subtilis* to the causative agent of anthrax, *B. anthracis*. Protein predictions were automatically integrated into a custom relational database with an associated Web portal to facilitate expert curation, results browsing and querying. We describe the use of workflow technology to arrange secreted proteins into families and the subsequent study of the relationships between and within these families. The design of the workflows, the architecture and the reasoning behind our approach are discussed.

1. Introduction

Bioinformatics has become one of the major applications areas for e-Science. The development of high-throughput technologies and the desire to understand organisms as complex systems, rather than the more traditional approach of studying their component parts, is placing new requirements for a computing infrastructure.

Outside of a few specialist centres, the majority of bioinformatics tasks lack the extreme requirements for raw computing power and large-scale storage, typical of physics. Moreover, bioinformatics tools and datasets tend to be highly heterogeneous in content and structure; the datasets are often widely geographically dispersed, and, as they are often maintained and deployed by individual scientists within their own laboratories, autonomously controlled. Dealing with these three problems – heterogeneity, distribution and autonomy – often forms a significant part of the work load of a bioinformatician.

Many bioinformatics experiments can be represented as a series of workflows, integrating a number of programs and data sources to test a hypothesis. These workflows, normally called "pipelines" within the field, form the bedrock of the computational analysis within bioinformatics. Traditionally, these have been implemented in two different ways. Firstly, in those laboratories with the necessary resources or specialist support, they have often been automated using Perl (which is ideally suited to the manipulation of the textual representations that biology has traditionally used to store its data). The majority of biologists, however, have used cut-and-paste between the myriad of Web sites offering access to underlying computational resources; in a sense, biology has predated the Web Service revolution, albeit in a rather *ad hoc* manner.

In previous work [1,2] we have described the [my]Grid project which aims to provide an alternative to these two approaches. The use of Web Services, a workflow enactment engine and a convenient, easy-to-use workflow editor, Taverna [3], have enabled lab biologists to access some of the power of automation available previously only to programmers.

In this paper, we describe the application of [my]Grid technology to an additional biological problem. We wish to understand and predict the characteristics and behavior of a family of bacteria, through an analysis of their complete genomic sequences. In this work we focus on a family of bacteria, *Bacillus*, whose members show a diverse range of properties. In particular, we wish to identify the proteins that are

produced by these bacteria and secreted across the cytoplasmic membrane.

This problem places different requirements on the workflow and the surrounding architecture than previously described in the bioinformatics workflow domain. Here, we describe in detail the background to the problem and the biological analysis that we wish to perform to address this problem. Finally, we describe the architecture that we have developed to support this analysis and discuss some preliminary results of its application.

2. The Secretome

One of the main mechanisms that bacteria use to interact with their environment is to synthesise proteins and export them from the cell into their external surroundings. These secreted proteins are often important in the adaptation of bacteria to a particular environment. The entire complement of secreted proteins is often referred to as the *secretome*. Characterising secreted proteins and the mechanisms of their secretion can reveal a great deal about the capabilities of an organism. For example, soil organisms secrete macromolecular hydrolases to recover nutrients from their immediate surroundings. During infection, many pathogenic bacteria secrete harmful enzymes and toxins into the extracellular environment. These secreted virulence proteins can subvert the host defence systems, for example by limiting the influence of the innate immune system, and facilitating the entry, movement and dissemination of bacteria within infected tissues [4]. Bacteria may also use secreted proteins to communicate with each other, allowing them to form complex communities and to adapt quickly to changing conditions [5].

The secretomes of pathogens are therefore of great interest. The comparison of virulent and non-virulent bacteria at the genomic and proteomic levels can aid our understanding of what makes a bacterium pathogenic and how it has evolved [6]. Furthermore, the characterisation of secretory virulence related proteins could ultimately lead to the identification of therapeutic targets.

The interest in protein secretion not only includes the secreted proteins themselves, but also those proteins which form the secretory machinery used to export the proteins across the cytoplasmic membrane, and in the case of Gram-negative bacteria, the outer membrane. Secretory proteins incorporate a short sequence called a signal peptide, which acts as a trafficking signal directing the protein to the secretory machinery. Not all proteins that are translocated from the site of synthesis are secreted into the surrounding medium; *transmembrane proteins* are localised to the cytoplasmic membrane itself; *lipoproteins* attach themselves to the outer surface of the membrane, protruding into the periplasmic space of Gram-negative bacteria or the membrane/wall interface of Gram-positive bacteria. Finally, proteins may also attach themselves to the cell wall by covalent or ionic interactions; the former may be distinguished by an LPXTG motif that is found in the C-terminal domains of mature secreted proteins [7].

Recently, our understanding of the secretome has greatly improved due to the rapidly increasing number of complete bacterial genome sequences, essentially molecular blueprints containing the information that allows the protein repertoire of an organism to be defined. Armed with this sequence information, we can begin to predict which of the proteins an organism is capable of producing are destined to be secreted, as well as the mechanisms of their secretion. As a result, a number of studies have been undertaken, in which bioinformatics programs and resources play a vital role. These studies involve the repeated application of a number of different algorithms to all of the gene and protein sequences encoded on the genome. Many of these algorithms are computationally expensive and, given that an average bacterial genome can encode around 4,000 or more proteins, the process can become computationally bound. In addition, the results of the application of these algorithms needs to be stored and integrated in order to make a prediction about the secretory status of the entire set of proteins encoded by a particular genome. Often, the results of the classification algorithms may be error prone and mechanisms to permit expert human curation and results browsing also need to be established.

Biologists have already begun to apply conventional bioinformatics technology to the prediction and classification of secreted proteins. The 'first, largely genome-based survey of a secretome' was carried out using bioinformatics tools on the genome of the industrially important bacterium, *Bacillus subtilis* [8], using legacy tools called from custom scripts in combination with expert curation.

In this study we describe the development and application of e-Science workflows and a service-oriented approach to the genomic scale detection and characterisation of secreted

proteins from *Bacillus* species. *Bacillus* species are important not only for their industrial importance in the production of enzymes and pharmaceuticals, but also because of the diversity of characteristics shown by the members of this genus. The *Bacillus* genus includes species that are soil inhabitants, able to promote plant growth and produce antibiotics. The genus also includes harmful bacteria such as *Bacillus anthracis*, the causative agent of anthrax.

We utilised the system to make predictions about the secretomes of 12 *Bacillus* species for which complete genomic sequences are publicly available; this includes *B. cereus* (strain ZK/E33L), *B. thuringiensis* konkukian (strain 97-27), *B. anthracis* (Sterne), *B. anthracis* (Ames ancestor), *B. anthracis* (Ames, isolate Porton), *B. cereus* (ATCC 10987), *B. cereus* (strain ATCC 14579/DSM 31), *B. subtilis* (168), *B. licheniformis* (strain DSM 13/ATCC 14580, sub_strain Novozymes), *B. licheniformis* (DSM 13/ATCC 14580, sub_strain Goettingen), *B. clausii* (KSM-K16) and *B. halodurans* (C-125/JCM 9153). These predictions were automatically integrated into a custom relational database with an associated Web portal to facilitate expert curation, results browsing and querying.

3. Workflow Approach

In this study, the aim was to identify proteins that are likely to be secreted and classify them according to the putative mechanism of their secretion. Such proteins include those exported to the extracellular medium, as well as proteins that attach themselves to the outer surface of the membrane (lipoproteins) and cell wall binding proteins (sortase mediated proteins containing an LPXTG motif). Once secreted proteins had been classified, we then investigated the composition of the predicted secretomes in the twelve species under study.

Two workflows were designed and implemented; the *classification* workflow and the *analysis* workflow. The classification workflow is concerned with making predictions about the secretory characteristics of a particular protein from a given set of proteins. The analysis workflow processes the data from the first workflow in order to analyse the function of the secreted proteins that have been found.

A general feature of the workflows is their linear construction. In the classification workflow, for example, at each step, the set is reduced in size, removing those proteins that do not require further classification. A conceptual diagram illustrating the functionality of the classification workflow is shown in Figure 1. The results of the classification process are stored in a remote relational database and the reduced set of proteins passed to the next service in the workflow. In the analysis workflow, data derived from the classification workflow is retrieved from the database and analysed using a further series of service enabled tools. The architectural constraints responsible for this choice of design are discussed further in section 4. The data flow for both workflows combined is summarised in Figure 2.

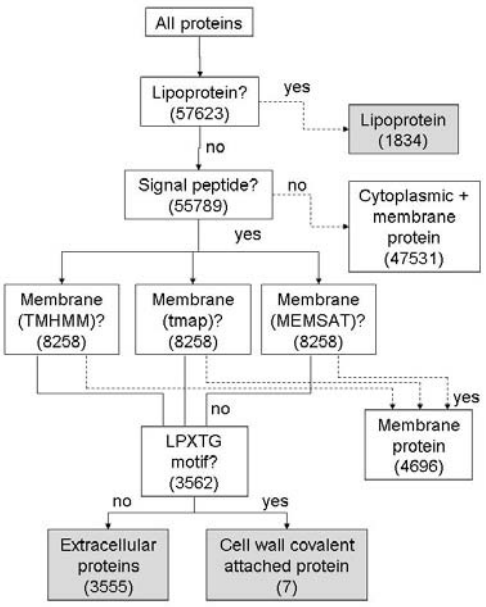


Figure 1. Basic representation of the functionality of the classification workflow. Shaded boxes indicate the set of secreted proteins. The number in brackets represents the total number of proteins to be classified at each level, across the 12 *Bacillus* species.

With respect to the workflow functionality, for the classification workflow, the first objective was the prediction of lipoproteins. This was the responsibility of the first service in the workflow, which takes as input a set of all predicted proteins derived from the EMBL record for the complete genome sequence. This service employed LipoP [9] for the detection of lipoproteins. Following the prediction of lipoproteins, the putative 'non-lipoprotein' set of proteins was analysed for the presence of a signal peptide using a SignalP Web Service [10]; this was performed after the lipoprotein identification because of possible limitations in the efficiency of SignalP at detecting

lipoproteins. The use of SignalP at this point also removes most of the proteins from subsequent analysis; this has considerable advantages as the downstream analyses are potentially computationally intensive.

Proteins with a putative N-terminal signal peptide as well as additional transmembrane domains are likely to be retained in the membrane. To identify these putative membrane proteins from among the proteins in the signal peptide dataset, we used a combination of three transmembrane prediction Web Services based on the tools, TMHMM, MEMSAT and tmap, respectively [11]. A subsequent service in the workflow was responsible for integrating the results derived from these three tools, to make a final prediction about the presence of a putative transmembrane protein. Finally, the protein dataset corresponding to proteins with no predicted transmembrane domain was analysed for the cell wall binding amino acid motif LPXTG. The tool, ps_scan (the standalone version of ScanProsite) which scans PROSITE for the LPXTG motif [12] was wrapped as a Web Service and called from the workflow. The classification workflow was validated in its predicted capability by applying it to the proteins of *Bacillus subtilis* whose secretory status has been determined and, to a large extent, experimentally confirmed [8, 13].

For the analysis workflow, the secreted proteins dataset was analysed to provide information about the relationships between the secretomes of the 12 different organisms in the study. Putative secreted proteins were extracted from the database, clustered into families, and the structure, functional composition and relationships between these families were studied. The set of secreted proteins includes those predicted to be lipoproteins, cell wall binding, or extracellular. Transmembrane proteins and cytoplasmic proteins were disregarded. Analysis of the data was initiated by clustering the putative secretory proteins into protein families using the MCL graph clustering algorithm [14]. MCL provides a computationally efficient means of clustering large datasets. BLASTp data was used with the MCL algorithm to identify close relatives of the predicted secreted proteins. This approach follows that of [15], where the BLASTp algorithm provides a score for the putative similarity between proteins. The necessary BLASTp data was retrieved from the Microbase system, a Web Service enabled resource for the comparative genomics of microorganisms [16]. For each predicted secreted protein, similar proteins with a BLASTp expect value less than $1e^{-10}$ were used as input to MCL, (inflation value 1.2).

Hierarchical clustering was performed to identify phylogenetic relations between the *Bacillus* species in the context of their contributions to the protein families. The R package was wrapped as a Web Service for this purpose. R is a package providing a statistical computing and graphics environment (R Project website, http://www.r-project.org/). A distance matrix was constructed using the Euclidean distance, and clustering was carried out using a complete linkage method.

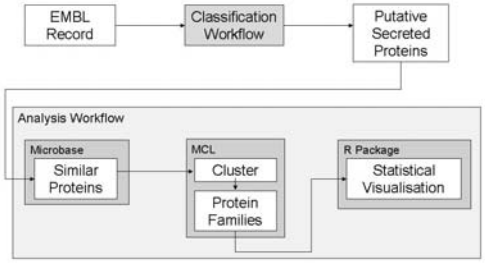


Figure 2. Data flow through the classification and analysis workflows.

4. Architecture

The architectural view of the workflow components involved in the prediction and analysis of the secretomes of the *Bacillus* species is shown in Figure 3.

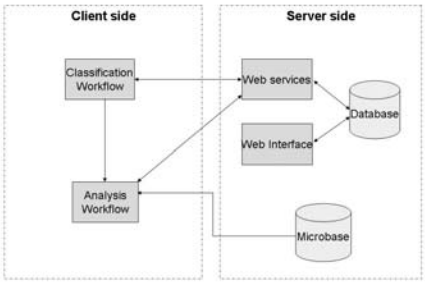


Figure 3. Architectural layout of the classification and analysis workflow components.

We sought to avoid implementing services on the client machine from which the workflows were enacted. This is particularly important for those services interacting with the database. In order to maximise performance, we endeavoured to locate the service as close to the database as possible. However, we also found continual problems with the reliability and availability of services provided by outside, autonomous individuals. Another problem was the restrictions placed on the usage and

dissemination of some tools. Running such tools locally required licensing agreements to be signed, limiting the tool usage to the licensees. As a result of these factors, many of the required services were implemented in-house, although still delivered using the Web Services infrastructure. The services were orchestrated using SCUFL workflows and constructed and enacted using the Taverna workbench [3].

The workflows implement the bioinformatics processes outlined in Figure 2. Most of the services both use and return text based information, in standard bioinformatics formats. At all steps of the classification workflow, we have extracted the intermediate results into a custom-designed database. This was achieved using a set of bespoke data storage services which parse the raw textual results, and store them in a structured form. It is these structured data which are used to feed later services in both the classification and analysis workflows. In our case, the custom database is hosted on a machine in close network proximity to the Web Services. This has the significant advantage of reducing the network costs involved in transferring data to and from the database.

After completion of the classification workflow, the custom database contains the data relating to each protein analysed, including the raw data, as well as an integrated summary of the analysis. Tracking the provenance of the data is important in this context, because there are a number of different routes for the classification workflow to designate a protein as 'secretory'. The basic operational provenance provided by Taverna also aids in the identification of service failures [17]. This is particularly important while running the transmembrane domain prediction services, as these run concurrently; a failure in one, therefore, does not impact on the execution of the classification workflow, although may return incomplete conclusions and thus needs to be recorded.

We have developed a Web portal to provide a user-friendly and familiar mechanism for accessing the secretomes data in the database.[1] From this site, users can select the bacterial species in which they are most interested and view the corresponding results. Data is initially displayed as a table from where the users can navigate further to view the details of the classifications. The protein sequences may be viewed along with an overlay of predicted signal peptides and their cleavage sites. Users may also edit and curate the database as appropriate. A screenshot of the database portal is shown in Figure 4.

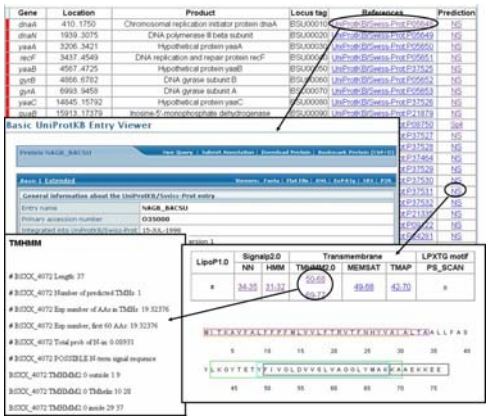


Figure 4. Screenshot of the Web portal summarising the characteristics of predicted secreted proteins from *B. subtilis* (168).

The analysis process is the most computationally intensive section requiring a large number of BLASTp searches. BLAST is the most commonly performed task in bioinformatics [1], and as such there are many available services which could have been used. However, because of the computational intensive nature of large BLAST searches, we retrieved pre-computed BLAST results from the Microbase database. Microbase is a Grid based system for the automatic comparative analysis of microbial genomes [16]. Amongst other data, Microbase stores all-against-all BLASTp searches for over three hundred microbial genome sequences, and the data are accessible in a Web Service-based fashion.

The final visualisation steps in the analysis workflow were performed using R and MCL. Again, both services were implemented locally, although both were exposed using Web Services. Completion of the entire workflow took approximately 2-3 hours.

This architecture differs from others using myGrid technology. Whilst the original requirements appeared to favour a highly distributed approach, we have found that the technological and licensing constraints have led to a hybrid approach: a combination of services and workflows, combined with databases for both results storage and pre-caching of analyses. The combination of all of these technologies does result in fairly complex architecture but provides a system that is fast and reliable enough for practical use.

[1] At http://bioinf.ncl.ac.uk:8081/BaSPP

5. Results

The classification workflow was applied to the predicted proteomes of the 12 *Bacillus* spp. listed in section 2. The resulting predicted secretomes varied in size from 358 proteins in *B. clausii* (strain KSM-K16) (9% of the total proteome) up to 508 proteins in *Bacillus cereus* (strain ZK / E33L) (11% of the total proteome). An investigation into the functional distribution of the proteins comprising the secretome of each strain was carried out by classifying them into functionally related families based on their sequence similarity as defined by BLASTp. The member proteins of the 12 secretomes were arranged into 543 families of 2 or more members. Some 350 proteins showed no similarity to other proteins and hence did not fall into families. Core protein families that contain members from all 12 proteomes are of particular interest since they may represent secreted proteins whose functions are indispensable for growth in all environments. 9% of protein families were found to be 'core' and the functions of these were investigated. The Gene Ontology terms [18] of the genes encoding the proteins in each cluster were examined and then summarised by classifying the terms according to the "SubtiList" classification codes [19]. Figure 5 shows a summary of the different functional classifications of the 'core' secreted protein families. Interestingly, a large number of core families had not been experimentally characterised and remain of unknown function. More predictably, many core proteins were grouped into families concerned with cell wall related functions, transporter proteins and proteins responsible for membrane biogenesis.

In addition to defining core protein families, we were also interested in gaining some insight into the functions of protein families that are specific to pathogens and those that are specific to non-pathogens. Canned queries over the database allowed these results to be easily and repeatedly extracted. Interestingly, only 14 of the 106 protein families that were unique to the secretomes of the potentially pathogenic bacteria (*B. cereus* (ZK / E33L), *B. thuringiensis* konkukian (97-27), *B. anthracis* (Sterne), *B. anthracis* (Ames ancestor), *B. anthracis* (Ames, isolate Porton), *B. cereus* (strain ATCC 10987) and *B. cereus* (ATCC 14579 / DSM 31)), showed similarity to proteins with known functions. Thus, the majority were found to be of unknown function and remain to be characterised.

The secretory protein families that were unique to the non-pathogens showed functions that were indicative of their habitats. Of the 11 unique families, 5 encoded enzymes concerned with the breakdown and transport of plant polysaccharides, 2 were concerned with the structure of flagellae, and the remaining 4 were functionally unclassified.

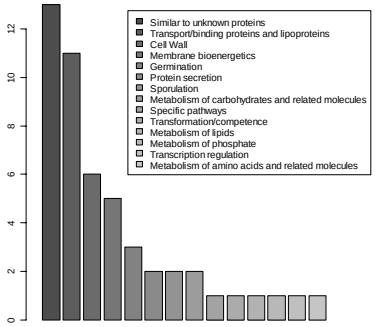


Figure 5. Functional classification of the 'core' secreted protein families. The graph shows the number of 'core' secreted proteins per 'SubtiList' category.

Finally, we were interested to determine whether the secretomes of the pathogenic organisms were closely related to each other in terms of their functional composition than to those of the non-pathogens. The phylogeny of the *Bacillus* strains was investigated in the context of the relationships between their secretomes. This was illustrated using a dendrogram, constructed using the R package, in which the relation between the different *Bacillus* strains is based on their contribution to the predicted secreted protein families (Figure 6). Essentially the dendrogram highlights the level of similarity between the secretomes of the various strains.

Within the *B. cereus* group subcluster (*B. anthracis*, *B. cereus* and *B. thuringiensis*), two sub-clusters were formed by the well-established pathogens (CP000001 *B. cereus*, AE017355 *B. thuringiensis* konkukian, AE017225 *B. anthracis*, AE017334 *B. anthracis*, AE016879 *B. anthracis)*, while the two members of questionable pathogenesis (AE017194 *B. cereus*, AE016877 *B. cereus)* formed a separate cluster. The environmental strains (*B. subtilis*, *B. licheniformis*, *B. clausii*, *B. halodurans*) formed a separate cluster from that of the *B. cereus* group organisms.

6. Discussion

***From the biologist's perspective*:** From the perspective of a biologist, construction of a workflow that enables secretory protein prediction over bacterial genomes using

multiple prediction tools, integrates the results into a database, and then performs analysis on the families, is a novel development. This approach utilises current bioinformatics programs in order to make predictions, which would otherwise take several days if performed manually. In particular the ease by which a workflow may be re-run as and when new genomes are sequenced is a distinct advantage, especially as the rate of complete genome sequencing continues to increase. The initial time in developing the application should also be considered; though this factor reduces in significance with re-execution of the workflow across a number of genomes.

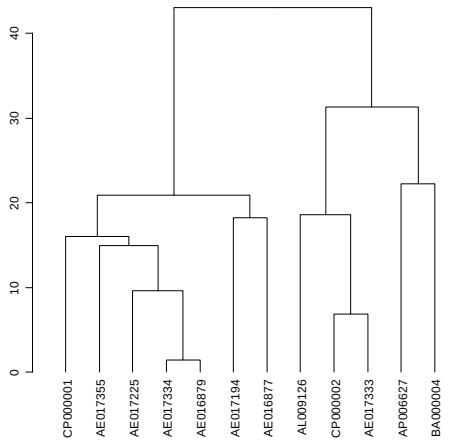


Figure 6. Dendrogram representing the relationship of the *Bacillus* species in terms of their secretome. From left to right: CP000001 *B. cereus* (ZK/E33L), AE017355 *B. thuringiensis* konkukian (97-27), AE017225 *B. anthracis* (Sterne), AE017334 *B. anthracis* (Ames ancestor), AE016879 *B. anthracis* (Ames, isolate Porton), AE017194 *B. cereus* (ATCC 10987), AE016877 *B. cereus* (ATCC 14579/DSM 31), AL009126 *B. subtilis* (168), CP000002 *B. licheniformis* (DSM 13/ATCC 14580, sub_strain Novozymes), AE173333 *B. licheniformis* (DSM 13/ATCC 14580, sub_strain Goettingen), AP006627 *B. clausii* (KSM-K16), BA000004 *B. halodurans* (C-125 / JCM 9153).

Another advantage for the biologist is that the generated data is stored in a custom database making the results available for subsequent analysis. It also provides a way of sharing data and promoting collaboration by providing an accessible and user-friendly Web interface to the database. This approach is different to current workflow methods where users need to take more control of where results are stored and possibly provide a means for the parsing of the data.

In addition to the bioinformatics approach, of course, the data generated by these workflows is also of great interest to microbiologists and this work provides data to prime further, biologically oriented investigations.

***From the eScientist's perspective*:** We have shown here the effectiveness of an e-Science approach, generating biologically meaningful results. These results can be used to generate hypotheses that may be verified by experimental approaches.

This process of data analysis was simplified through the use of Microbase. The comparison of the bacterial proteins had already been done, therefore reducing the computing time required in analysing the results.

Although the details of the workflow are specific to the problem of predicting secretory proteins, the architectural solutions that we have employed represent general issues for e-Science. We have attempted to deal with three key problems – distribution, autonomy and heterogeneity, in an efficient manner. While the e-Science framework has helped, it has been less successful in dealing with some key issues.

Dealing firstly with the problems of **autonomy**; most of the services in the classification workflow are, in fact, provided originally by external parties, autonomous from the workflow authors. As has been mentioned by previous authors [1], the **reliability** of the services deployed by many providers is not high. The problem has been significantly worsened in this case, as the workflows are largely linear; therefore, a failure by any service will cause the entire workflow to fail. We solved this problem by the simple approach of hosting the services locally, which is obviously not ideal. Having developed local services, we would have liked to at least republish them for reuse as a service to the community. There is, however, the second problem of **licensing** agreements: in most cases, we are not allowed to expose services for use by non-licensed individuals.

Perhaps surprisingly, there were few problems introduced by **distribution** in this work. The main recurrent difficulty came from the relatively large datasets over which we were operating. This was one of the motivations for the linear shape of the classification workflow. We can see two, more principled approaches to this problem. Firstly, **improved data transport** facilities, enabling transfer without SOAP packaging, as well as direct transfer between third parties, would reduce many difficulties. The new Taverna2 architecture should enable these functionalities [20]. Secondly, the ability to **migrate workflows** and services closer to the

data would provide a significant advantage; in many cases the service executables are smaller than the data they operate over. It should be noted that licensing issues will prevent this in many cases.

Data **heterogeneity** has provided us with fewer problems than expected for a bioinformatics workflow. There are relatively few data types, most often protein lists are passed between the services in the workflows. Data heterogeneity was dealt with through the use of a custom database and parsing code. In the second workflow, the use of the Microbase data warehouse removed many of these problems.

One major future enhancement is to use, notification in conjunction with the workflow. This would provide a way of automatically analysing recently annotated genomes for secretory proteins. The need for users to interact with the workflow would therefore be removed, providing automatic updates of the database with new secretory protein data. We also intend to investigate the migration of workflows from the machine of their conception, to a remote enclosed environment from which they are able to access services that are unable to be exposed directly to third parties.

In conclusion, the investigation of predicted bacterial secretomes through the use of workflows and e-Science technology indicates the potential use of computing resources for maximising the information gained as multiple genomic sequences are generated. The knowledge gained from large scale analysis of secretomes can be used to generate inferences about bacterial evolution and niche adaptation, lead to hypotheses to inform future experimental studies and possibly identify proteins as candidate drug targets.

7. Acknowledgments

We gratefully acknowledge the support of the North East regional e-Science centre and the European Commission (LSHC-CT-2004-503468). We thank the EPSRC and Non-Linear Dynamics for supporting Tracy Craddock.

8. References

1. Stevens RD, Tipney HJ, Wroe CJ, Oinn TM, Senger M, Lord PW, Goble CA, Brass A, Tassabehji M., 2004. *Bioinformatics*, 20 Suppl. 1:I303-I310.
2. Li P, Hayward K, Jennings C, Owen K, Oinn T, Stevens R, Pearce S and Wipat A. *Proceedings of the UK e-Science All Hands Meeting 2004, 31st Aug - 3rd Sept, Nottingham UK*.
3. Oinn T, Addis M, Ferris J, Marvin D, Senger M, Greenwood M, Carver T, Glover K, Pocock MR, Wipat A, Li P, 2004. *Bioinformatics*, 20(17): 3045-54.
4. Nomura K, He SY, 2005. *PNAS*. 102(10): 3527-3528.
5. Piazza F, Tortosa P, Dubnau D, 1999. *J Bacteriol*., 181(15): 4540-8.
6. Lee VT, Schneewind O, 2001. *Genes Dev.*, 15(14): 1725–1752.
7. Boekhorst J, de Been MW, Kleerebezem M, Siezen RJ, 2005. *J Bacteriol*., 187(14): 4928-34.
8. Tjalsma H, Bolhuis A, Jongbloed JDH, Bron S, van Dijl J.M, 2000. *Microbiol. Mol. Biol. Rev.*, 64(3): 515–547.
9. Juncker AS, Willenbrock H, von Heijne G, Nielsen H, Brunak S, Krogh A, 2003. *Protein Sci*., 12(8): 1652-62.
10. Nielsen, H. & Krogh., A., 1998. *Proc Int Conf Intell Syst Mol Biol. (ISMB 6)*, *6*: 122-130.
11. Moller S, Croning MDR, Apweiler R, 2001. *Bioinformatics*, 17(7): 646-653.
12. Gattiker A, Gasteiger E, Bairoch A, 2002. *Appl Bioinformatics*, 1(2): 107-108.
13. Tjalsma H, Antelmann H, Jongbloed JDH, Braun PG, Darmon E, Dorenbos R, Dubois JY, Westers H, Zanen G, Quax WJ, Kuipers OP, Bron S, Hecker M, van Dijl, J.M., 2004. *Microbiol. Mol. Biol. Rev.*, 68: 207-233.
14. van Dongen S, 2000. A cluster algorithm for graphs. *Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam*.
15. Enright AJ, Van Dongen S, Ouzounis CA, 2002. *Nucleic Acids Res.*, 30(7):1575-1584.
16. Sun Y, Wipat A, Pocock M, Lee PA, Watson P, Flanagan K, Worthington JT, 2005. *The 5th IEEE International Symposium on Cluster Computing and the Grid* (CCGrid 2005), Cardiff, UK, May 9-12.
17. Zhao J, Stevens R, Wroe C, Greenwood M. Goble C, 2004. In Proceedings of the *UK e-Science All Hands Meeting, Nottingham UK, 31 Aug-3 Sept.*
18. Gene Ontology Consortium, 2006. *Nucleic Acids Res.*, 34 (Database issue):D322-6.
19. Moszer I, Jones LM, Moreira S, Fabry C, Danchin A, 2002. *Nucleic Acids Res.*, 30(1): 62-5.
20. Oinn T & Pocock M, pers. comm

Proceedings of the UK e-Science All Hands Meeting 2006, © NeSC 2006, ISBN 0-9553988-0-0

Distributed Analysis in the ATLAS Experiment

K. Harrison[a], R.W.L. Jones[b], D. Liko[c], **C. L. Tan**[d]

[a]Cavendish Laboratory, University of Cambridge, CB3 0HE, UK
[b]Department of Physics, University of Lancaster, LA1 4YB, UK
[c]CERN, CH-1211 Geneva 23, Switzerland
[d]School of Physics and Astronomy, University of Birmingham, B15 2TT, UK

Abstract

The ATLAS experiment, based at the Large Hadron Collider at CERN, Geneva, is currently developing a grid-based distributed system capable of supporting it's data analysis activities which require the processing of data volumes of the order of petabytes per year. The distributed analysis system aims to bring the power of computation on a global scale to thousands of physicists by enabling them to easily tap into the vast computing resource of various grids such as LCG, gLite, OSG and Nordugrid, for their analysis activities whilst shielding them from the complexities of the grid environment. This paper outlines the ATLAS distributed analysis model, the ATLAS data management system, and the multi-pronged ATLAS strategy for distributed analysis in a heterogeneous grid environment. Various frontend clients and backend submission systems will be discussed before concluding with a status update of the system.

1. Introduction

Based in the European Laboratory for Particle Physics (CERN) [1], Geneva, the ATLAS experiment [2] is set to commence its investigations into proton-proton interactions at the most powerful particle accelerator in the world, the Large Hadron Collider (LHC) [3], in the summer of 2007. Each high energy proton-proton collision will produce hundreds of particles in the detector. Highly efficient software and electronics filter and record interactions of potential physics interest for subsequent analysis. The estimated annual yield is 10^9, corresponding to around 10 petabytes of data.

ATLAS involves a collaboration of more than 2000 physicists and computer scientists from over 150 universities and laboratories in 35 countries across 6 continents. In anticipation of the unprecedented volume of data that will be generated by the ATLAS detector when data taking commences, and large-scale simulation, reconstruction and analysis activities, particle physicists have adopted Grid technology to provide the hardware and software infrastructure required to facilitate the distribution of data and the pooling of computing and storage resources between world-wide collaborating institutions. The ATLAS grid infrastructure currently consists of three grids: LCG (LHC Computing Grid) [4], OSG (US-based Open Science Grid) [5] and Nordugrid (grid project based in the Nordic countries) [6].

The primary aim of the distributed analysis project is to bring the power of computation on a global scale to individual ATLAS physicists by enabling them to easily tap into the vast computing resource provided by the various grids for their analysis activities whilst shielding them from the complexities of the grid environment.

This paper begins by outlining the ATLAS distributed analysis model [7]. It goes on to describe the data management system adopted by ATLAS, then details the ATLAS strategy for distributed analysis [8] in a heterogeneous grid environment. Various frontend clients will be introduced followed by descriptions of different submission systems and their associated grid infrastructure. The paper concludes by providing a status update of the ATLAS distributed analysis system.

2. Distributed Analysis Model

The distributed analysis model is based on the ATLAS computing model [7] which stipulates that data is distributed in various computing facilities and user jobs are in turn routed based on the availability of relevant data.

A typical analysis job consists of a Python [9] script that configures and executes a user-defined algorithm in Athena (the ATLAS software framework) [10] with input data from a file containing a collection of potentially interesting particle interaction information or events and producing one or more files containing plots and histograms of the results.

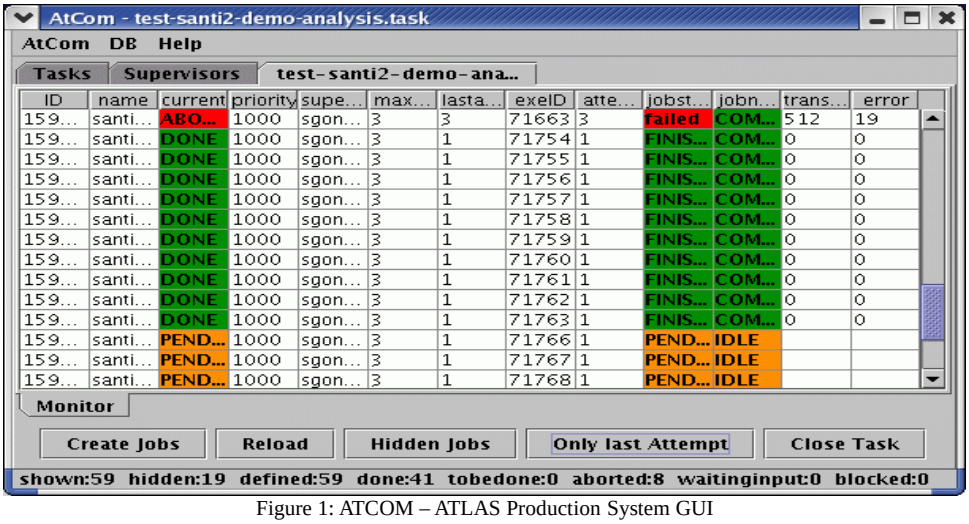


Figure 1: ATCOM – ATLAS Production System GUI

As with many large collaborations, different ATLAS physics groups have different work models and the distributed analysis system needs to be the flexible enough to support all current and newly emerging work models whilst remaining robust.

3. Distributed Data Management

With data distributed at computing facilities around the world, an efficient system to manage access to this data is crucially important for effective distributed analysis.

Users performing data analysis need a *random access* mechanism to allow rapid pre-filtering of data based on certain selection criteria so as to identify data of specific interest. This data then needs to be readily accessible by the processing system.

Analysis jobs produce large amounts of data. Users need to be able to store and gain access to their data in the grid environment. In the grid environment where data is not centrally known, an automated management system that has the concept of file ownership and user quota management is essential.

To meet these requirements, the Distributed Data Management (DDM) system [11] has been developed. It provides a set of services to move data between grid-enabled computing facilities whilst maintaining a series of databases to track these data movements. The vast amount of data is also grouped into *datasets* based on various criteria (e.g. physics characteristics, production batch run, etc.) for more efficient query and retrieval. DDM consists of three components: a *central dataset catalogue*, a *subscription service* and a set of *client tools* for dataset lookup and replication.

The central dataset catalogue is in effect a collection of independent internal services and catalogues that collectively function as a single *dataset bookkeeping system*.

Subscription services enable data to be automatically *pulled* to a site. A user can ensure that he/she is working on the latest version of a dataset by simply subscribing to it. Any subsequent changes to this dataset (i.e. additional files, version changes, etc.) will trigger a fresh download of the updated version automatically.

Client tools provide users with the means to interact with the central dataset catalogue. Typical actions include listing, retrieving and inserting of datasets.

4. Distributed Analysis Strategy

ATLAS takes a multi-pronged approach to distributed analysis by exploiting its existing grid infrastructure directly via the various supported grid flavours and indirectly via the ATLAS Production System [12].

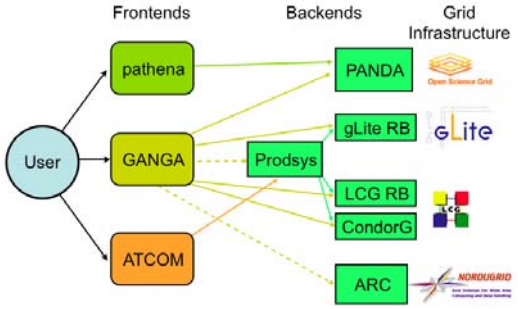


Figure 2: Distributed analysis strategy

4.1 Frontend Clients

Figure 2 shows various frontend clients enabling distributed analysis on existing grid infrastructure.

Pathena [13] is a Python script designed to enable access to OSG resources via the Panda job management system [14]. It is just short of becoming a drop-in replacement for the executable used in the ATLAS software framework. Users are able to exploit distributed resources for their analysis activities with the very minimal inconvenience.

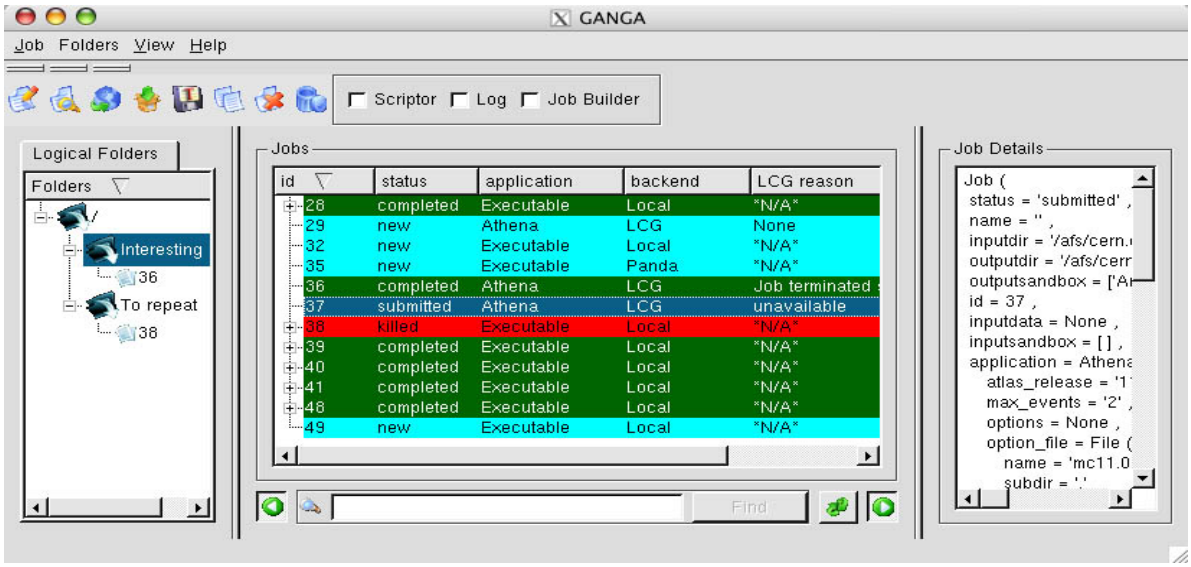


Figure 3: GANGA - Job definition and management tool

Pathena makes the submission of analysis jobs to the Panda system a painless two stage process involving an optional *build* step (where user code can be compiled) followed by an *execution* step (with built-in job splitting capabilities). A further *merge* step is in development which will allow the resulting output datasets from split jobs to be consolidated.

ATCOM [15] is the dedicated graphical user interface frontend (See Figure 1) to the ATLAS production system designed to be used by a handful of expert users involved in large-scale organised production of ATLAS data. It has potential to be used for end-user distributed analysis purposes.

GANGA [16] is a powerful yet user friendly frontend tool for job definition and management, jointly developed by the ATLAS and LHCb [17] experiments. GANGA provides distributed analysis users with access to all grid infrastructure supported by ATLAS. It does so by interfacing to an increasing array of submission backend mechanisms. Submission to the production system and ARC are planned for the not so distant future.

GANGA currently provides two user interface clients: a Command Line Interface (CLI) and a Graphical User Interface (GUI) (See Figure 3). In addition, it can also be embedded in scripts for non-interactive/ repetitive use. GANGA, due to its need to satisfy ATLAS and LHCb experiment requirements (unlike Pathena and ATCOM which are specialised tools designed for specific ATLAS tasks), has been designed from the onset to be a highly extensible generic tool with a component plug-in architecture. This pluggable framework makes the addition of new applications and backends an easy task.

A synergy of GANGA and DIANE [18] (a job-distribution framework) has been adopted in several instances. In each instance, GANGA was used to submit various types of jobs to the Grid including the search for drugs to combat Avian flu, regression testing of Geant 4 [19] to detect simulation result deviations and the optimisation of the evolving plan for radio frequency sharing between 120 countries.

A few physics experiments (e.g. BaBar [20], NA48 [21]) have also used GANGA in varying degrees while there are others (e.g. PhenoGrid [22], Compass [23]) in the preliminary stages of looking to exploit GANGA for their applications.

GANGA is currently in active development with frequent software releases and it has an increasing pool of active developers.

4.2 Production System

The ATLAS production system provides an interface layer on top of the various grid middleware used in ATLAS. There is increased robustness as the distributed analysis system benefits from the production system's experience with the grid and its *retry and fallback* mechanism for both data and workload management.

A rapidly maturing product, the production system provides various facilities that are useful for distributed analysis e.g. user configurable jobs, X509 certificate-based access control and a native graphical user interface, ATCOM.

4.3 LCG and gLite

LCG is the computing grid designed to cater to the needs of all the LHC experiments and is by default the main ATLAS distributed analysis target system. Access to the LHC grid resources is via the LCG Resource Broker (RB) or CondorG [24]. The LCG RB is a robust submission mechanism that is scalable, reliable and has a high job throughput. CondorG, although conceptually similar to the LCG RB,

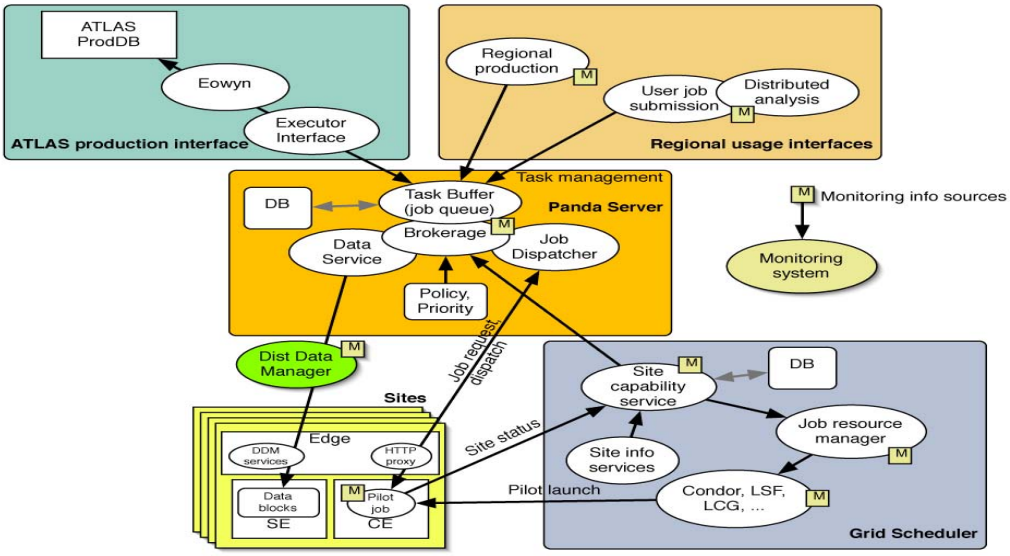


Figure 4: Panda architecture

has a different architecture. Nevertheless, both submission mechanisms have been successfully exploited in recent large-scale ATLAS production exercises.

gLite [25] is the next generation grid middleware infrastructure project by EGEE (Enabling Grids for E-SciencE) [26]. Recent code base convergence between gLite and LCG has resulted in gLite 3.0. The gLite RB has performance enhancements that are of particular benefit to distributed analysis: efficient bulk job submissions with improved support for output result retrieval.

GANGA supports direct analysis job submission to the CondorG, LCG RB and the gLite RB.

4.4 Panda

Panda is a job management system associated with OSG designed specifically for both distributed production and distributed analysis. See Figure 4.

Panda has native support for the ATLAS DDM system allowing it to accept DDM datasets as input (pre-staging it where required) and producing DDM datasets as output (retrievable using DDM tools).

Panda offers users a comprehensive system view which presents heterogeneous distributed resources as a single uniform resource accessible though a standard interface. It also has extensive web-based job monitoring and browsing capabilities.

Panda does not have a GUI but it looks to GANGA to provide it with a graphical job definition and submission interface. Panda has made this possible by exposing a useful set of client API.

4.5 ARC

ARC (Advanced Resource Connector) [27] is developed by the Nordugrid collaboration and is based on the Globus Toolkit [28].

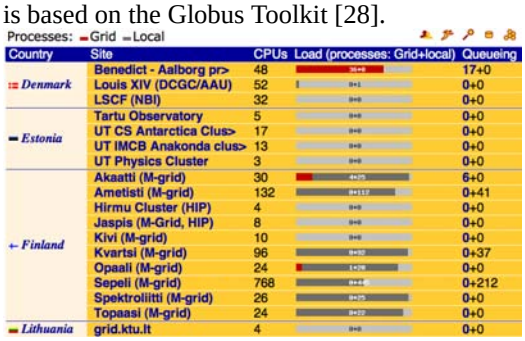


Figure 5: Nordugrid Grid Monitor

The ARC client is a light-weight, self-contained job submission tool with built-in highly customisable resource brokering functionality and input / output file staging facilities. It has an impressive job delivery rate of approximately 30-50 job deliveries/min making it potentially useful for interactive (i.e. responsive) distributed analysis.

As with all other grid flavours, the ARC client has a comprehensive set of command line tools for job submission and management. The web-based Nordugrid Grid Monitor [29] complements the ARC client by providing detailed system-wide job monitoring information for *all* jobs running on Nordugrid resources.

Although not specifically designed for distributed analysis, ARC has immense potential due to it's stability and performance. Certain issues with data management still need to be finalised. GANGA is looking to interface with Nordugrid in the not too distant future.

5. Conclusion

Distributed analysis in ATLAS is still in its infancy but is evolving rapidly. Many key components like the DDM system have only just come online. The multi-pronged approach to distributed analysis will encourage one submission system to learn from another and ultimately produce a more robust and feature-rich distributed analysis system. The distributed analysis system will be comprehensively tested and benchmarked as part of Service Challenge 4 [30] in the summer of 2006.

Acknowledgements

We are pleased to acknowledge support for the work on the ATLAS distributed analysis system from GridPP in the UK and from the ARDA group at CERN. GridPP is funded by the UK Particle Physics and Astronomy Research Council (PPARC). ARDA is part of the EGEE project, funded by the European Union under contract number INFSO-RI-508833.

References

[1] http://cern.ch
[2] ATLAS Collaboration, Atlas - Technical Proposal, CERN/LHCC94-43 (1994); http://atlas.web.cern.ch/Atlas/
[3] LHC Study Group, The LHC conceptual design report, CERN/AC/95-05 (1995); http://lhc.web.cern.ch/lhc/
[4] http://lcg.web.cern.ch/LCG/
[5] http://www.opensciencegrid.org/
[6] http://www.nordugrid.org/
[7] ATLAS Computing Group, ATLAS Computing Technical Design Report, CERN-LHCC-2005-022; http://atlas-proj-computing-tdr.web.cern.ch/atlas-proj-computing-tdr/PDF/Computing-TDR-final-July04.pdf
[8] D. Liko et al., The ATLAS strategy for Distributed Analysis in several Grid infrastructures, in: Proc. 2006 Conference for Computing in High Energy and Nuclear Physics, (Mumbai, India, 2006); http://indico.cern.ch/contributionDisplay.py?contribId=263&sessionId=9&confId=048
[9] G.van Rossum and F.L. Drake, Jr. (eds.), Python Reference Manual, Release~2.4.3 (Python Software Foundation, 2006); http://www.python.org/
[10] http://cern.ch/atlas-proj-computing-tdr/Html/Computing-TDR-21.htm#pgfId-1019542
[11] ATLAS Database and Data Management Project; http://atlas.web.cern.ch/Atlas/GROUPS/DATABASE/project/ddm/
[12] ATLAS Production System; http://uimon.cern.ch/twiki/bin/view/Atlas/ProdSys
[13] T. Maeno, Distributed Analysis on Panda; http://uimon.cern.ch/twiki/bin/view/Atlas/DAonPanda
[14] T. Wenaus, Kaushik De et al, Panda - Production and Distributed Analysis; http://twiki.cern.ch/twiki//bin/view/Atlas/Panda
[15] http://uimon.cern.ch/twiki/bin/view/Atlas/AtCom
[16] http://ganga.web.cern.ch/ganga/
[17] LHCb Collaboration, LHCb - Technical Proposal, CERN/LHCC98-4 (1998); http://lhcb.web.cern.ch/lhcb/
[18] http://it-proj-diane.web.cern.ch/it-proj-diane/
[19] http://geant4.web.cern.ch/geant4/
[20] http://www-public.slac.stanford.edu/babar/
[21] http://na48.web.cern.ch/NA48/
[22] http://www.phenogrid.dur.ac.uk/
[23] http://wwwcompass.cern.ch/
[24] http://www.cs.wisc.edu/condor/condorg/
[25] http://glite.web.cern.ch/glite/
[26] http://egee-intranet.web.cern.ch/egee-intranet/gateway.html
[27] http://www.nordugrid.org/middleware/
[28] http://www.globus.org/
[29] http://www.nordugrid.org/monitor/
[30] LHC Computing Grid Deployment Schedule 2006-08, CERN-LCG-PEB-2005-05; http://lcg.web.cern.ch/LCG/PEB/Planning/deployment/Grid%20Deployment%20Schedule.htm

Proceedings of the UK e-Science All Hands Meeting 2006

Nottingham, UK

18th – 21st September

Author Index