ISSN 1751-5971

UK Experience with OGSA

Report of the workshop held at CCLRC, May 28th-29th, 2003

Dave Berry

Research Manager, National e-Science Centre

for the workshop participants:

Neil Chue-Hong (EPCC)
Rob Allan, Xiao Dong Wang, Daniel Hanlon (CCLRC)
Mark McKeown (Grid Support Centre)
Guy Rixon (Cambridge)
Savas Parastatidis (Newcastle)
Soren Sorensen, Stefano Street (UCL)
Don Cruickshank (Southampton)
Terry Harmer (Belfast)
Bing Wu, Ivaylo Kostadinov (Oxford)
Steven Newhouse, William Lee (London)
Dave Berry (NeSC)

National e-Science Centre

This report is available online at
http://www.nesc.ac.uk/technical_papers/UKeS-2003-02.pdf
Electronic copies of presentations made at this workshop are available at
http://wiki.nesc.ac.uk/read/gt3-users?MeetingMay03

Abstract

This document summarises a meeting that was held at the CCLRC's "Cosener's House" conference centre in May 2003. This meeting brought together people from a range of UK e-Science projects who are using the OGSA architecture, in particular as implemented by the alpha release of Globus Toolkit version 3. The delegates were chosen to reflect a range of projects, from applications to middleware.

Each delegate gave a presentation about their project, and the discussion identified common problems and requirements. The meeting produced several outcomes, including this report, and a mini-workshop at the UK e-Science All-Hands Meeting in September 2003. In particular, the meeting produced a list of OGSA services and functionality required by these projects, ordered by priority.

Introduction

OGSA[1] and OGSI[2] are major standards in Grid computing, and several UK e-Science projects are either using them or planning to use them. However, at the time of writing the specification and implementation of OGSI was still fluid, with the Globus Toolkit 3 (GT3)[3] still in alpha release. For this reason, the UK e-Science Grid is currently relying on the more stable Globus Toolkit 2 (GT2), which does not implement OGSI. Until this meeting, there has not been a forum for UK OGSA projects to share experiences.

This meeting provided such as forum. It brought together people from a number of UK OGSA projects. Each delegate gave a presentation about their project, and the discussion identified common problems and requirements.

The delegates represented a range of projects, from applications to middleware. However, the list of projects was not exhaustive, and the deliverables from the meeting should be taken as indicative rather than authoritative.

The projects all use Globus Toolkit 3, which is the main reference implementation of OGSI. At the time of the meeting, this was in alpha release. Furthermore, the specification of OGSI was changing as it converged on its first release. So the technology was immature, and the projects were very much in the role of early adopters, with the frustrations as well as excitement that this always generates. The meeting attempted to identify the key issues and requirements involved in expanding the use of OGSA in UK e-Science.

The meeting aimed to produce the following deliverables:

1. A plan and abstract for a mini-workshop at the UK e-Science All-Hands Meeting,
2. Pointers to FAQs and HOWTOs,
3. A list of services required by application projects,
4. A list of resources produced in the UK,
5. A brief description of current issues and concerns, and
6. This report of the meeting.

The following sections present the list of services, list of resources, and issues. These are followed by a description of each project represented at the meeting. The appendices give the delegates at the meeting, the abstract for the mini-workshop, and a Wiki set up for sharing information between users of GT3.

Requirements

This section describes the items that the delegates felt necessary to better support the use of OGSA in the UK e-Science programme.

Services

This section lists the services that our projects need to implement their required functionality. It presents these services in a series of stages, starting with the absolutely minimal set that are required to implement any applications at all, then identifying the key services for most projects, and moving on to discuss services for Information Grids and Compute/File Grids. Finally, it lists the functionality that projects will require from the Grid infrastructure, as opposed to the services required to build the actual applications.

This list complements the Gap Analysis report[4] produced by Geoffrey Fox and David Walker for the UK e-Science Programme. Their Gap Analysis provides a broad picture of what is required for a fully functional Grid. By contrast, this list focuses on the core services that application projects believe they will need, ordered by urgency.

The list does not distinguish services by where they are deployed. Some may be deployed globally, some on each machine, and some in a mixture. For example, a project may require data access services on one machine and job submission services on another.

The delegates assumed that the UK will use OGSA services when they become available. This list is not an attempt to define an alternative to the OGSA. It is just a list of services we thought would be beneficial for the UK grid. When appropriate OGSA services exist, the UK Grid should adopt those.

Absolutely Minimal Set

This set loosely corresponds to the services that are currently available. Given that the meeting consisted of people who are currently using GT3, this outcome is not surprising. The main addition is the simple registry service.

Simple Registry. This should be built from OGSI Service Groups, and provide simple functionality for use within a single application.

GridFTP/Reliable File Transfer. An OGSI port of the GT2 utility.

Job Submission. Again, an OGSI update of the GT2 functionality.

Data Access. As provided by the current release of OGSI.

Authentication. Identification of users via digital certificates, and access control based on GridMap files or similar.

Key services/2nd stage

These services are those required by almost every application. With these services in place, a wide range of applications could be supported.

Authorisation Services. These should provide more detailed access control than GridMap files. They could be implemented using systems such as PERMIS, AKENTI or Shibboleth.

Notification. This should build upon the notification system in OGSI.

Workflow. The UK e-Science Gap Analysis has identified this as a major missing piece of functionality, and several UK e-Science projects have developed workflow packages. This service should integrate with both OGSA-DAI services and compute services.

Dependency management. The projects represented at this meeting are already encountering problems with clashing dependencies. A deployed Grid must include functionality for managing dependences between services, and also dependencies between services and the underlying computing fabric.

Registry. A more sophisticated registry than the simple service above, this would probably be implemented using UDDIe[5] or R-GMA[6].[*]

Information Grid services

Information Grids are those that integrate large-scale distributed data repositories. They require services that go beyond the simple data access services in the absolutely minimal service set.

Distributed Query Processing. This is the ability to query data stored in multiple places. Ideally this service would allow different data formats to be queried in a single query, but earlier versions are more likely to support only relational queries or only XML queries.

Data Cache. This service will store data in an accessible form or location, to improve performance for repeated accesses.

Data Replication. Data replication services already exist, but need to be brought into the OGSA framework.

Mobile Code. Often it is much more efficient to move the computation to the data than vice versa. This will require support for deploying code on a data resource.

Compute/File Grid services

Compute/File Grids are largely aimed at supporting traditional scientific jobs such as particle physics data analysis.

Schedulers. The terminology of scheduling, brokering and workload management is not consistent. This entry covers a range of such services, in particular the automated distribution of compute jobs across compute nodes.

Dynamic deployment of applications. Compute/File Grids in particular require the deployment of applications on compute servers. Currently this is mostly done by hand, and is very time-consuming. This activity is a subset of the broader notion of *provisioning* of resources and services.

[*] ISI are also developing a registry capability for GT3, which includes support for “soft state” entries.

Portal services

Portals require particular services to manage the group of services presented to a user.

Session management. This maintains the identity of a user session across a set of service interactions.

State management. This maintains that state of a set of services in a session, communicating state from one service to another as required.

Other services

Clearly there are many other services that can be built on top of a functioning Grid. This section lists some of those of particular interest to the delegates at the meeting.

Reservation. For complex workflows, it will be necessary to reserve and co-allocate distributed resources so that they can work together

Transaction. A related notion is that of a transaction. A transaction is a group of actions in a workflow such that either all happen or none of them happen. In practice this is implemented using rollbacks. The commercial world is built on transactions, and they apply to the scientific world as well.

Dynamic deployment of services. As services are updated, and new services are created, projects will need to deploy these remotely. This topic is similar to "Dynamic deployment of applications" mentioned above, and as more applications are implemented as services, the distinction between the two will fade.

Data Mining. It may be possible to create a framework for implementing a range of data mining services.

Visualisation. Similarly, it may be possible to create a framework for implementing a range of visualisation services.

Infrastructure

The previous section listed the services and functionality that projects require to build their applications. This section lists the functionality that projects will require from an OGSA Grid infrastructure. The functionality is presented as related groups, but no attempt has been made to attach priorities to the different functionalities.

Service Deployment

Easy deployment. Several delegates reported that deploying services was too hard. More work is needed to simplify the packaging and deployment of user services.

Service Versioning. A mechanism is required for identifying the version of a deployed service.

Life Cycle Management. Deploying a service is not a one-time operation. New versions of the service will be released, and old services will be deprecated. Mechanisms are needed for managing this process. (This is not to be confused with managing the life-cycles of service instances, which OGSI provides some support for).

Central Registry

Applications can implement their own registries, but a Grid infrastructure will require a central registry. This should list the ***Service Factories*** and ***Data Sets*** available on the Grid.

User Management

Community Authorisation. A mechanism is required to simplify the authorisation of particular users for a given project. Existing work includes VOM[7], VOMS[8] and CAS[9].

Account generation. When a user is newly authorised to use a given resource, this often requires that a new account is created on that resource. This is time consuming. A degree of automation would simplify the operation of the Grid.

Monitoring

Integration Tests. The UK e-Science Engineering Task Force[10] has found that good integration tests are essential. These scripts check the connectivity of every pair of sites on the Grid, and report failures so that the administrators can investigate problems.

Resource Usage. The administrators of each site will require the ability to monitor the use of their resources.

Logging. Here, logging refers to logging the use of Grid resources and traffic between them. It should be noted that a logging service will also be useful to applications for debugging and analysis.

Accounting. Resources have to be paid for, and users need to know how much their computation will cost.

Documentation

Some projects are producing documentation to help projects to get to grips with using OGSA and GT3. The meeting felt that much more is required. In particular, we would like to see explanations of how to write certain kinds of services. Most people were keen that documentation should include sample code, and vice versa, i.e. all sample code should be accompanied by clear explanatory text.

Summary

The delegates identified the key services that their projects required to exploit an OGSA Grid. In contrast with the UK Gap Analysis, they ordered the services by priority. The resulting list shows that there is a large amount of work to be done, even without attempting the breadth of work listed in the Gap Analysis.

A useful activity would be to compare the list given here with other documents, such as the Gap Analysis and the OGSA specification. It may well be that some requirements were missed by the meeting – for example, the delegates did not list computational steering among the list of desired services, nor programming models or debugging, let alone Semantic Grids.

Available Resources

This section briefly describes some resources produced by the projects represented at the meeting.

Documentation and Examples

Some of the delegates mentioned documentation resources that they could contribute to the community.

Members of the OGSA-DAI[11] project have written several pages of notes to guide people through the installation of GT3. Belfast e-Science Centre[12] have written several "How to" notes, covering issues such as configuring Axis[13] and Tomcat[14], writing a first service, and writing a secure service.

NeSC have created a Wiki for GT3 users (see the appendices for details). It was suggested that these notes could be added to the Wiki, so that the UK community could find them easily.

NeSC developed a course on Grid Services, in conjunction with Stilo[15]. A reduced version of this course should be released under the Globus licence. This could also be added to the Wiki.

The meeting noted that we should liaise with the Globus project, the European Grid Support Centre[16], and of course the UK Grid Support Centre[17]. They may have existing material that would be useful; conversely they may be interested in the notes that we write ourselves.

New Services from projects

Several projects have produced middleware services that could be of use to other projects.

OGSA-DAI have produced services for accessing databases:

- GridDataService
- GridDataServiceFactory
- GridDataServiceRegistry
- GridDataTransport
- Distributed Query Processing

MyGrid[18] have produced services for co-ordinating systems:

- Workflow (based upon WSFL)
- Notification

The NEReSC Core Middleware[19] project will package the above services into a single tested package.

The IeSE[20] project is planning to produce some portal services, such as session management.

In addition to middleware services, some projects have produced services which may be useful samples for others to follow. AstroGrid[21] has a prototype database service, and the 3D OPT Microscopy Grid[22] has implemented a Grid Service I/F to 3D visual reconstruction code.

Other new functionality

AstroGrid can provide three tools or techniques that might be useful for other projects. The first of these is a Java template for creating a simple service. This simplifies that task of creating a service, provided that the service implements a single port type with no service data. (This hasn't been tested with Globus Toolkit 3.0).

The second technique is for packaging Grid Services as WAR files, in such a way that services running in the same instance of Tomcat can use different versions of OGSA. This is accomplished by extending the Ant build files to include all the necessary library files in the WAR files. This allows you to add services to Tomcat, and remove them, without having to restart Tomcat.

The third tool from AstroGrid is a delegate class for hiding instance creation. This simplifies the creation of client programs for existing services, and is also well suited to be called from Java Server Pages.

Anyone interested in the AstroGrid utilities should contact Guy Rixon (gtr@ast.cam.ac.uk).

Issues Raised

This section lists various issues and concerns that the delegates raised.

GT3 Platform

As the meeting was specifically convened to bring together users of GT3, it is unsurprising that many of the issues raised concern that toolkit. As previously noted, at the time of the meeting, this was in alpha release. The concerns of the delegates in part reflected the pre-release nature of the software.

JAR versioning

Several projects encountered problems with clashing versions of JAR files. For example, a JAR used by or distributed with GT3 would clash with another version of the same file required by a different package. This is not an easy problem to solve. It indicates that if GT3 is to be deployed across the UK, a support body should test compatibility between different packages, to reduce the burden on application projects.

Performance

Several projects raised concerns about the performance of GT3. This requires some investigation. It is not an easy problem, as the performance may depend on many other systems, including the JVM, the Web Server, the Service container, the operating system, the services used, and external factors such as the network.

Documentation

All projects reported major gaps in the GT3 documentation. Some have written help notes on how to deploy the system, but these only address some of the gaps in the system. Security is one area where we considered the documentation to be opaque, and the non-core services (see next item) also need more documentation.

Non-core services are hard to build and deploy

In general, the core OGSI services in GT3 can be deployed easily. The surrounding services, such as job submission and file transfer, are much more opaque. This is partly because these services are currently transitional (they wrap GT2 services rather than implement OGSA services), and partly because the documentation is poor.

Port to IPv6

The 6net[23] project ported GT3 to run of IPv6. This required some changes to the implementation. As IPv6 may be required for some UK projects, this support should be built-in to the OGSA implementation.

Some problems with Windows path names

Some projects reported problems using the GT3 core services on Windows, when path names contained spaces or were too long.

Singleton services

Some projects encountered problems trying to create singleton services. These are services that have only one instance, as opposed to services that are created by Factories. OGSI includes a notion of "persistent" services that are supposed to provide this functionality, but these projects couldn't get these to work. It's not clear whether this is a problem in the implementation, the documentation, or the client code.

Container-managed persistence

Some projects wanted the service container to have the ability to save the state of OGSA services across container restarts. This would make the services robust across system crashes and similar problems. GT3 does not yet provide this.

GT3 WSDL doesn't show bindings

This is a more philosophical issue about the level of abstraction in WSDL. A conventional (non-GT3) WSDL interface contains the port type and a set of bindings. The port type describes the interface of the service, while the bindings describe the location of the service and the protocols used to access it. This has the problem that it is not possible to describe a service interface without specifying the actual service that implements it.

GT3 supports abstraction by allowing WSDL files to omit bindings. However, this means that the interface does not describe the protocols required (e.g. SOAP/HTTP, CORBA/IIOP). This is a deliberate design decision. However, not all protocols work together, and some delegates believe that it may be necessary to specify the protocols in WSDL.[*]

Other considerations

Some questions were raised about the deployment of GT3 services. For example, since front-end machines will be running web service hosting environments, will they need to be more powerful machines than with GT2? Also, since when a container crashes, all contained applications are lost, is GT3 less robust than GT2? In general, delegates thought that it should be easier to deploy services than it is now.

Although the meeting concentrated on GT3, there was some discussion about support for other platforms, such as .NET[24]. One concern was how interoperable these different implementations of OGSI would be in practice. Currently all implementations use the GT3-provided bindings, but in experiments the University of Virginia OGSI.NET[25] team have found that Java and .NET implementations returned subtly different XML data from some calls.

---

[*] This has since been discussed in the OGSI mailing list and at GGF8. Although not clearly stated in the OGSI specification, it is assumed that the basic WS-I profile is used for HTTP/SOAP interoperability. If a service assumes another binding (e.g., CORBA) then nothing is specified by the specification. Another issue raised during an OGSI meeting in GGF8 is the lack of any language in the OGSI specification in relation to the transformation of SDE's to WSDL 1.1. This is going to be addressed by a document that is currently under development.

On a related note, some delegates wondered how we could check services for compliance with OGSA. A test suite or test bed may be useful in this regard.* This is related to the problem of version tracking and life-cycle support, because it seems likely that the definition of OGSA will change over time.

Security is an ever-present issue with the Grid. One question is whether system administrators will see GT3 as more or less of a threat than GT2. On the one hand, web services are possibly more secure than running jobs natively; on the other hand, web services make it easier for people to deploy and use the Grid. Delegates also suggested that the Grid community ought to put more effort into evaluating intrusion detection systems, rather than only relying on authentication.

Summary

The meeting raised several concerns with the current state of GT3. No doubt some of these will be resolved as the Globus team respond to feedback, but others are worth further investigation. The concerns about performance are the most worrying for the long term.

* As of GGF8, there are no plans at GGF level for an OGSA interoperability suite, nor even an OGSI interoperability suite.

Reports from GT3-based Projects

This section briefly summarises the reports from each project represented at the meeting. The emphasis of the reports in on the projects' use of GT3.

OGSA-DAI (Neil Chue-Hong[*])

OGSA-DAI has produced an implementation of the Grid Data Services Specification (GDSS), as defined by the GGF Data Access and Integration Services working group (DAIS-WG[26]). It is also working on other areas of data access and integration, most notably Distributed Query Processing (DQP)[27].

The GDSS defines two extra port types: GridDataService & GridDataTransport. It also defines two ways to expose activities: a document-based one and an RPC-like one. GridDataService represents a data source. GridDataTransport represents the movement of data between different services. There is an ongoing discussion regarding GridDataTransport, involving the DAIS WG, the Globus team, and others, to see whether this port type can be made a more general and pervasive part of the OGSA.

The OGSA-DAI implementation is in Java. It follows the GDSS, ignoring RPC interfaces for now. It implements three main services: GridDataServiceRegistry, GridDataServiceFactory, and GridDataService.

The implementation has been tracking GT3 since GT3 Technology Preview 4. At the beginning it was very difficult to get things done, but this has improved over time. The developers found the GT3 Core easy to use, compared to other EPCC projects that have had problems with other GT3 services such as Job Submission and Security. Recent changes to OGSI have been major, e.g. changing the registry port type to service groups, and more work was required to track them.

There remain lots of "gotchas" in using GT3. For example, the developers have 4 pages of notes about possible installation problems. They have also encountered problems with different versions of JAR files, e.g. when GT3 requires one version and another package requires a different version. On Windows, they found some problems with long path names and path names with spaces. The OGSI developers list has been some help, but this is not intended for most users. The documentation and tooling improved with the release of GT3 Alpha 4.

There are many design and implementation issues remaining. On the design side, the DAIS WG is discussing how to expose DBMS notifications via OGSA interfaces. There is also continuing discussion over how to expose activities, i.e. Document vs. RPC. It is an open question how persistent services should be handled. On the implementation side, the team have stayed shy of security issues so far. They also have concerns whether the OGSI Core Service container is efficient enough.

The ODD-Genes[28] project is a small demonstrator that mixes GT3 (OGSA-DAI) and GT2 (SunDCG[29]).

[*] EPCC, University of Edinburgh

MS.NETGrid (Neil Chue-Hong)

MS.NETGrid[30] is producing an implementation of OGSI on .NET, and a set of training material to be used in courses to be given at the e-Science Institute. The implementation will leverage as many .NET capabilities as possible, both to minimise development work and to make the system fit naturally into the .NET environment. The system will consist of an ASP.NET web application and a set of service libraries.

The University of Virginia are also working on an implementation of OGSI under .NET (called OGSI.NET). The two projects discuss design with each other. They have slightly different requirements; for example, MS.NETGrid is working to a tight time scale, whereas OGSI.NET is concerned with building a more robust system. MS.NETGrid has little reliance on GT3 except for WSDL files and cross-implementation testing.

EQUATOR-MIAS (Don Cruickshank[*])

The EQUATOR-MIAS[31] project is part of the EQUATOR IRC[32]. It focuses on wearable devices for medical health. These devices provide live data collection, such as accelerometers, ECGs, thermometers, GPS locators, and others. These can be used to detect medical problems as they arise, and the system can send recommendations to the patient, such as reminders to take readings, or warnings to take medication after detecting features in their sensor data.

The Grid architecture uses proxies for the sensors, a database (OGSA-DAI), a feature detector, and the clinician's interface. It uses ServiceData and notifications to send events, and will be adapted to make use of Service Groups. So far the team have shied away from security, although they recognise that they will need to address this.

The overall architecture, if you abstract over the details of the processing services, is quite general, and is also being used in an Antarctic Remote Sensing project. In that project the readings are values such as thickness of the ice, depth of water, and light level under the ice.

One complication introduced by the scenario is that it requires real-time configuration as devices are added and removed from the Grid. This may include a new type of sensor, in which case the system must dynamically create database tabled to store the readings from sensors of that type. Configuration messages may take several forms, including SOAP, SMS and e-mail.

One question that arose during discussion was whether GT3 notifications are secure?

NEReSC Core Middleware (Savas Parastatidis[†])

This project arose when the NEReSC team noticed several commonalities in the software requirements for their projects. They investigated the possibility of re-using software components, with the aims of reducing testing and training.

---

[*] University of Southampton

[†] University of Newcastle

NEReSC will produce a package of four widely-used services:

- OGSA-DAI GDSS
- OGSA-DAI DQP
- MyGrid Workflow
- MyGrid Notification

This team will perform quality control, testing the individual components of the package and interoperability between them, to ensure that the versions of the services work together as they should. They will provide user documentation and test programs. The package will be available for download.

NEReSC will create tutorial material that makes use of the Core Middleware package. One aspect of the tutorial material is likely to be guidance on converting web services to OGSA, as this is a common requirement among a number of e-Science projects.

In the future, the team would like to track changes to the services, making the package OGSA-compliant throughout. They plan to include new services, developing some if necessary. They would like to see these services deployed as standard around the UK, with a common registry of long-lived services.

e-Protein (Soren Sorensen, Stefano Street[*])

The e-Protein[33] project is producing a prototype for national proteomic grid. The system submits jobs using a package called GenTHREADER, which executes the job using a Java wrapper. The original implementation used GT2 GRAM; they are now working on a GT3 prototype (although GT3 still uses GT2 GRAM for job submission).

A web interface can show the progress of computation of distributed nodes.

In future work, they plan to write a virtualised job submission environment that can hide the details of the underlying architecture from the programmer, and a virtual execution machine to interact with schedulers, process migration, etc.

6net (Soren Sorensen, Stefano Street)

6net is a project to port GT3 to support IPv6, and to build an ipv6-enabled Grid infrastructure. This required changes to be made to GT3 client and server code where IPv4 calls are currently made. A more detailed analysis to identify changes to non-web-based services (e.g. GridFTP).

The 6net and e-Protein projects encountered a small number of problems when installing GT3. These mainly affected 6net, which requires Java 1.4 for IPv6 support. This requires some changes to the GT3 code, and caused some JAR versioning conflicts with Tomcat. It also required a patched version of JDBC. Moving to GT3 from GT2 required some modifications to mapfiles, and other small changes, to handle security. The team added the Xalan.jar file to the installation, also to support security.

[*] University College London

AstroGrid (Guy Rixon)*

AstroGrid is the UK Virtual Observatory project. It plans to provide co-ordinated access to on-line archives, remote access to data processing services, Grid-accessible storage of results, the ability for end-users to publish data on the Grid, and for astronomers to share "private" data across sites. It will be primarily a data and service Grid, making use of remote processing, workflow, astronomy-specific services, File transfer, and GridFTP.

So far the team has built small prototypes to test the technology. An echo service tests GT3, including authentication tests, and a simple data selection service is testing OGSA-DAI. Currently they are working on a demonstration for the IAU conference. They do not plan to use GT3/OGSI in AstroGrid products, as opposed to prototypes, until 1Q2004.

In general, AstroGrid have found the GT3 Core reliable, and much better than GT2, but they have been limited by the poor documentation. They have not yet exploited some features of GT3, including service data, service groups, and notification. They are concerned by apparent poor performance, and the size of the deployed software. They wonder whether a lightweight client implementation would be possible.

AstroGrid like that split between factories and instances, but raise the question of how can programmers create singleton services (i.e. ones without factories)? They also aren't sure how to build registries using GT3.

They report that they found the build system opaque and poorly designed, and reported that it didn't always work. They tried building in separate tree, which worked. They were able to use a template for simple services. However, they found services hard to deploy. So they tried packaging each service with OGSI in a .war file. This eased deployment, but took some work to get running correctly.

The GT3 non-core services were too hard to build, on Solaris. GridFTP was very hard to set up, and they also encountered problems with firewalls.

Regarding security, AstroGrid are happy with the authentication systems in GT3. Authorisation is lacking: they want community-based authorization. They also want to use GSI with web portals. These are being addresses by various other projects in the UK and elsewhere, but are not yet ready for use.

IeSE (Rob Allan, Xiao Dong Wang, Daniel Hanlon†)

The IeSE project is building an Integrated Software Environment. It incorporates a range of Web Services into a portal: Data Discovery, Data Visualisation, Experiment Control and a Computing Grid.

The DataPortal[34] provides session management, role-based authorization, database triggers & customization front-ends to data service. It also includes a metadata catalogue, database search, and database download with a shopping cart service.

* University of Cambridge

† CCLRC, Daresbury

The HPCPortal[35] provides session management, using MDS[36], GRAM[37], GridFTP[38], GASS[39] and GRIP[40].

IeSE are writing some portal services, which sit above the basic services. These manage the integrated state of a whole session. They are using UDDI as a resource directory.

Most of this work has been done using web services. They are just starting with GT3.

ICENI (Steven Newhouse, William Lee[*])

ICENI[41] is a middleware system. Key to the design is an abstraction layer that separates the service publishing API from the service discovery API. The implementation is pluggable: the first implementation in Jini[42], and a student working on a JXTA[43] version.

The ICENI OGSA gateway links OGSA services to ICENI. It listens to GT3 service events, and when a service required it generates the necessary stubs, and deploys it to the container using the GT3 Programmable Service Deployment API, which can deploy ICENI services in a GT3 Container without having to restart Tomcat.

This work shows that it is possible to extend GT3 to bridge to other technologies. The system maps ICENI metadata to and from OGSI service data, and ICENI subcontracting to the OGSI Factory port type. This works without changes to the GT3 build and deployment system.

The ICENI OGSA gateway has tracked GT3 from the Technology Preview to alpha 3. The team found it difficult to keep pace with the GT3 development cycle. This has not been helped by the poor documentation. They find the performance poor; it isn't clear whether this is in Axis, the Servlet engine, the HTTP server, the GSI authentication in CoG, or another part of the system.

The team think that the "Operation Provider" model in GT3 alpha 4 improves over previous "delegation" model. But now they can't derive WSDL from Java – until perhaps the tooling is brought up to date?

BioSimGrid (Bing Wu[†])

BioSimGrid[44] is a Biological Simulation Database. It links results from a synchrotron to a compute Grid and metadata databases. The Grid core uses OGSA-DAI, GridFTP and Akenti for authorisation.

The database itself is about 2TB in size, and intended to be distributed between different hosts. It has two levels of metadata: the first describes simulation data, and the second describes results of generic analysis. It will be associated with software tools for interrogation and data mining, while the compute Grid will run generic analysis tools compatible with different simulation codes and platforms. The system as a whole will make use of web services for Data mining, Data Analysis, Visualisation, DQP, Service Monitor, Session Manager, Portal/Proxy, and Transactions.

---

[*] Imperial College of Science and Technology

[†] University of Oxford

When the system is running, a major concern of will curation of the data. The system will need to support annotation of simulation data, to track its provenance.

The project has several concerns with GT3 technology. It is not yet clear whether the performance will be satisfactory. They have not resolved all the security issues. They need support for transactions. Currently GT3 and OGSA-DAI are changing rapidly, which does not provide a stable base for development. Finally, there is a question whether web services will be sufficient for the project, in which case OGSA will be overkill; and if OGSA is necessary, what is the migration path between the two?

Oxford eSC (Ivaylo Kostadinov[*])

Ivaylo presented his experience as a user first installing GT3 Technology Preview 3 (TP3) and GT3 Alpha 3. TP3 installed fine, bar two filenames misconfigured in build.xml. Alpha 3 had some more problems. ldconfig had to be used to set up *.so paths, and one SGL file needed for the file transfer service was missing.

Ivaylo also presented Oxford's work on firewalls. As part of their contribution to the ETF, they have created a central database of approved systems. This can be used by system administrators to open ports as necessary.

GridCast (Terry Harmer[†])

The GridCast[45] project explores the use of the Grid in broadcast media. The initial focus is in using a Grid architecture to distribute programmes between broadcast sites. Television programmes are held as digital video files, and are very large – 25Gb per hour in broadcast form and 100-250 Gb per hour in production form. Currently, television programmes are distributed to BBC NI, BBC Scotland, and BBC Wales from BBC Network (London) using dedicated leased ATM circuits.

GridCast has written operational GT3 network services (internally) to experiment with service creation and interaction. Services will be deployed in BBC NI. The main interests of the project are in data replication, where they are using GT3 services to wrap the GT2 functionality, in scheduled movement of data files, in wrapping broadcast infrastructure services as grid services, and in resource discovery.

In the future the project aims to explore the use of a Grid architecture to monitor and facilitate programme production, and to facilitate access to broadcasting resources in television programme production.

The project has been slowed by the lack of documentation for GT3, particularly in the area of resource discovery and registration. They have written some "how to" documents on some other aspects of G£, such as how to configure Axis and Tomcat. They would like to see a UK repository of such documents, including well tested examples.

---

[*] University of Oxford

[†] Queen's University Belfast

eSI and NeSC (Dave Berry)

The e-Science Institute[46] has run workshops about GT3, and a series of courses earlier this year that covered Java, Web Services, and Grid Services. These courses used a commercial training company (Stilo) and were very well received, but were too expensive to continue at the same frequency. However, the IPR of the Grid Services course is held jointly by Stilo and NeSC, and a cut-down version will be released under an open source licence.

eSI will be appointing a training team later this year, funded by JISC, the EU, and PPARC. This will teach existing courses and develop new material. We may also be presenting a course in conjunction with the European Data Grid[47].

eSI encountered problems getting GT3 security to work in the examples for this course. The documentation was poor, the Core distribution was missing some scripts, and we had to go back to GT2 to create test certificates. The example didn't work in time, and needs to be revisited before releasing or re-running the course.

NeSC[48] is just starting a project funded by IBM, to evaluate IBM's Globus-based technology. For the first stage, we will write a sample application running a data mining program that detects satellites in images of the sky, so that they are not confused with the astronomical data of interest. NeSC is the only non-US site involved in this programme, although they may be interested in involving other UK projects later.

NeSC is also planning a GT3-based Grid to link the compute resources at the Universities of Edinburgh and Glasgow. The implementation of this is awaiting people with time that can be allocated to it, but the preparatory work has identified several issues that must be addressed, including user portals, account management, monitoring, application deployment, version management and a registry. These are similar to the issues being addressed by the UK e-Science Grid.

Appendix 1: The All-Hands Abstract

OGSA: The platform for delivering UK e-Science

Mini-workshop and panel

Dave Berry, NeSC
Steve Newhouse, LeSC
Rob Allan, CLRC
and invited panellists

Presentation 1: Experience with GT3 (15 mins)

Presentation 2: The Road Ahead for UK Grid Services (15 mins)

Panellist Presentations (5 mins per panellist)

Open Discussion (30-45 minutes)

This workshop will present and discuss the use of the OGSA platform in UK e-Science. A major part of the workshop will be a panel discussion, with questions from the floor.

Two talks will set the scene for the panel discussion. The first will summarise the experience of a number of projects that are already using OGSA, primarily in the form of Globus Toolkit 3. This summary will be taken from the report of a meeting in late May 2003, augmented by further experience in the time running up to the All-Hands Meeting. It will include a list of grid services and related functionality required in the short term.

The second talk will present a roadmap for adopting OGSA services in the UK. The intention is that this roadmap will reflect the thinking of the UK e-Science Directorate.

Following the two main presentations, the remaining panellists will be given 5 minutes/3 slides to present personal viewpoints of the topic. Then the discussion will be opened to the floor.

We plan to have between 4 and 6 panellists. Currently, Dave Berry is nominated to chair the meeting, but we may invite a senior member of the UK e-Science community to take on this role. The panellists will be invited from a range of backgrounds, such as the Globus Team, UK projects making use of OGSA, the Architecture Task Force, and Grid Service Providers. Some possibilities, in addition to the authors, include: Guy Rixon (AstroGrid), Savas Parastatidis (NEResc), Geoffrey Fox (Indiana), Simon Laws (IBM/OGSA_DAI), Lisa Childers (Argonne), Wolfgang Emmerich (UCL), and UK e-Science Directors.

Appendix 2: gt3-users Wiki

NeSC have created a Wiki for storing FAQs, HOWTOs, and general discussion about OGSA and GT3. The URL is **http://wiki.nesc.ac.uk/read/gt3-users**. Anybody can read this Wiki, and registered users can edit it.

To edit the Wiki, create an account at **http://support.nesc.ac.uk/resources/**. This requires you to enter a PIN, which can be obtained by contacting NeSC or another user of the Wiki. Once you have created an account, request edit permission from support@nesc.ac.uk.

References

[1] I.Foster, C.Kesselman, J.Nick, S.Tuecke, "The Physiology of the Grid", http://www.globus.org/research/papers/ogsa.pdf

[2] S.Tuecke, K.Czajkowski, I.Foster, J.Frey, S.Graham, C.Kesselman, T.Maquire, T.Sandholm, D.Snelling, P.Vanderbilt, "Open Grid Services Infrastructure (OGSI) Version 1.0", https://forge.gridforum.org/projects/ogsi-wg/document/Final_OGSI_Specification_V1.0/en/1/Final_OGSI_Specification_V1.0.pdf

[3] Globus Toolkits: http://www-unix.globus.org/toolkit/

[4] Gap Analysis Report: http://www.nesc.ac.uk/technical_papers/UKeS-2003-01/GapAnalysis30June03.pdf, and http://www.nesc.ac.uk/technical_papers/UKeS-2003-01/Appendix30June03.pdf

[5] UDDIe: http://www.cs.cf.ac.uk/user/A.Shaikhali/uddie/index.htm

[6] R-GMA: http://marianne.in2p3.fr/datagrid/documentation/rgma-guide.pdf

[7] VOM: http://www.lesc.ic.ac.uk/projects/oscar-g.html

[8] VOMS: http://grid-auth.infn.it/docs/VOMS-v1_1.pdf

[9] CAS: http://www.lesc.ic.ac.uk/projects/cas.html

[10] UK e-Science Engineering Task Force: http://www.grid-support.ac.uk/etf

[11] OGSA-DAI: http://www.ogsadai.org.uk

[12] Belfast e-Science Centre: http://www.qub.ac.uk/escience

[13] Apache AXIS: http://ws.apache.org/axis

[14] Tomcat: http://jakarta.apache.org/tomcat

[15] Stilo: http://www.stilo.com

[16] EU Grid Support Centre: http://www.grid-support.net

[17] UK Grid Support Centre: http://www.grid-support.ac.uk

[18] MyGrid: http://www.mygrid.org.uk

[19] NEReSC Core Middleware: http://www.neresc.ac.uk/projects/CGM/index.html

[20] IeSE: http://esc.dl.ac.uk/IeSE/about.html

[21] AstroGrid: http://www.astrogrid.org

[22] 3D OPT Microscopy Grid: http://www.gridoutreach.org.uk/docs/pilots/3dopt.htm

[23] 6net: http://www.6net.org

[24] .NET: http://www.microsoft.com/net/

[25] OGSI.NET: http://www.cs.virginia.edu/~humphrey/GCG/ogsi.net.html.

[26] DAIS-WG: http://forge.gridforum.org/projects/dais-wg

[27] DQP: http://www.neresc.ac.uk/projects/OGSA-DQP/index.html

[28] ODD-Genes: http://www.epcc.ed.ac.uk/~oddgenes/

[29] SunDCG: http://www.epcc.ed.ac.uk/sungrid/

[30] MS.NETGrid: http://www.epcc.ed.ac.uk/~ogsanet/.

[31] EQUATOR-MIAS: http://www.cs.ucl.ac.uk/research/equator/projects/escience

[32] EQUATOR IRC: http://www.equator.ac.uk

[33] e-Protein: http://www.e-protein.org/res_intro.html

[34] DataPortal: http://esc.dl.ac.uk:9000/index.html

[35] HPCPortal: http://esc.dl.ac.uk/HPCPortal/hpcportal.shtml

[36] MDS: http://www.globus.org/mds/mds2/

[37] GRAM: http://www.globus.org/gt2/GRAM.html

[38] GridFTP: http://www.globus.org/datagrid/deliverables/C2WPdraft3.pdf

[39] GASS: http://www.globus.org/gass/

[40] GRIP: http://www.grid-interoperability.org/

[41] ICENI: http://www.lesc.ic.ac.uk/iceni

[42] Jini: http://www.jini.org

[43] JXTA: http://www.jxta.org

[44] BioSimGrid: http://www.biosimgrid.org

[45] GridCast: http://www.qub.ac.uk/escience/projects.php#gridcast

[46] e-Science Institute: http://www.nesc.ac.uk/esi

[47] European DataGrid: http://eu-datagrid.web.cern.ch/eu-datagrid

[48] National e-science Centre: http://www.nesc.ac.uk