Experiences with Migrating ^{my}Grid Web Services to Grid Services

Savas Parastatidis (Savas.Parastatidis@newcastle.ac.uk)
Paul Watson (Paul.Watson@newcastle.ac.uk)

North-East Regional e-Science Centre
School of Computing Science, University of Newcastle upon Tyne
Newcastle upon Tyne, NE1 7RU, UK
http://www.neresc.ac.uk

The North-East Regional e-Science Centre is involved in a large number of research projects that rely on the design and development of a Grid-based infrastructure. Building a large number of different infrastructures would be time-consuming, difficult and risky; currently available OGSA-based middleware is relatively immature and subject to change, while the knowledge and experience of it among e-Science researchers is understandably very limited. Consequently, it was decided to analyse the requirements of all our projects and to design and build a common Core Grid Middleware consisting of a selected number of Grid Services. This required the migration of Web Services to Grid Services – a common requirement as many UK e-Science research projects have built their infrastructure on Web Services and are now considering a move to Grid Services. Therefore, this paper discusses the experience gained by migrating two Web Services to the Core Grid Middleware and making them OGSI-compliant.

1. Introduction

The North-East Regional e-Science Centre is involved in a large number of research projects that rely on the design and development of a Grid-based infrastructure. Building a large number of different infrastructures would be time-consuming, difficult and risky; currently available Grid middleware is relatively immature and subject to change, while the knowledge and experience of it among e-Science researchers is understandably very limited. A further problem is that many e-Science projects—both at Newcastle and elsewhere—are currently based on Web Services, but consideration is being given to moving to Grid Services at some point in the future. The method of making the transition is not yet clear, and nor is the extent of the extra work, delays and risks that this will introduce in projects.

Consequently, it was decided to analyse the requirements of all the Newcastle projects and to design and build a common *Core Grid Middleware* [1]. The core consists of a set of key Grid Services. Each project can adopt the Core Grid Middleware and build its own application-specific services on top of it. This should reduce development effort, time and risk.

Two of the initial services in the Core Grid Middleware had been initially developed as Web Services as part of the ^{my}Grid research project [2]. This document outlines the design and implementation decisions made during the process of converting these services to OGSI-compliant Grid Services.

The rest of this document is structured as follows. Section 2 describes the current state of the Open Grid Services Architecture and its implementation. Section 3 gives a brief overview of the NEReSC projects. Section 4 describes the initial composition of the Core Grid Middleware and the process by which it will be built and extended. Section 5 reports on the experiences from converting the ^{my}Grid Web Services to Grid Services. Finally, Section 6 describes additional future options and the conclusions of this investigation.

2. Grid Services

The Open Grid Standard Architecture (OGSA) [3] is the Grid community's effort to create a set of standards for the construction of interoperable and platform-neutral Grid applications. It will define a number of key Grid Services on which Grid applications will be built (Figure 1). Version 1.0 of the Open Grid Services Infrastructure

(OGSI) [4], the foundation of OGSA, has now been finalised.

OGSI defines the fundamental properties/characteristics of a *Grid Service* using the Web Services Description Language (WSDL) [5]. "*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*" [4]. The Globus project [6] recently released v3.0 of their Globus Toolkit [7], which, amongst other features, provides a reference implementation of the OGSI standard [4]. The toolkit also includes a number of tools for consumers and developers of Grid Services.

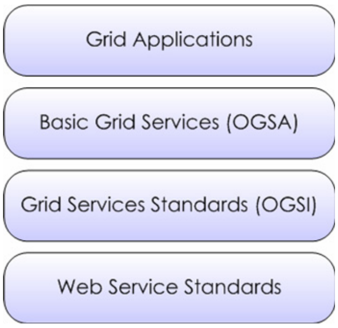

Figure 1: Grid application stack

The OGSA working group [8] is currently considering a number of use cases [3] in order to identify those core Grid Services that are going to make up the OGSA layer of Figure 1. A number of Grid Services that may find their way in the OGSA layer are presented in Figure 2. Global Grid Forum [9] working groups have already started the standardisation process for some of these services.

3. Current NEReSC Research Projects

The North-East Regional e-Science Centre (NEReSC) is involved in a number of Grid-related research projects: ^{my}Grid [2], OGSA-Data Access and Integration (OGSA-DAI) [10], Microbase [11], OGSA Distributed Query Processing (OGSA-DQP) [12], BASIS [13], e-Demand [14], GridMIST [15], GridSHED [16], eXSys [17], GOLD [18]. The promised deliverables of these projects include Grid middleware, Grid application frameworks for specific domains, and complete Grid applications.

Although researchers involved in these projects are already collaborating, there is a need to identify and build a common infrastructure, and provide the necessary development tools to assist them in their implementation work. This will prevent duplication of development effort and, as a result, avoid wasting valuable resources and time. Furthermore, it is important for the deliverables to be interoperable with existing and future Grid Service standards - currently most NEReSC projects are focused on Web Service standards (Figure 3) and so a migration path to Grid Services will be required (this is also true for many UK e-Science research projects).

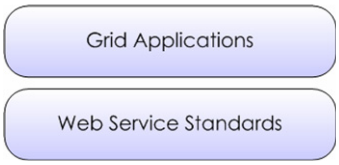

Figure 3: Application stack currently adopted by most of the NEReSC projects

Notable exceptions amongst the NEReSC research projects are OGSA-DAI and OGSA-DQP

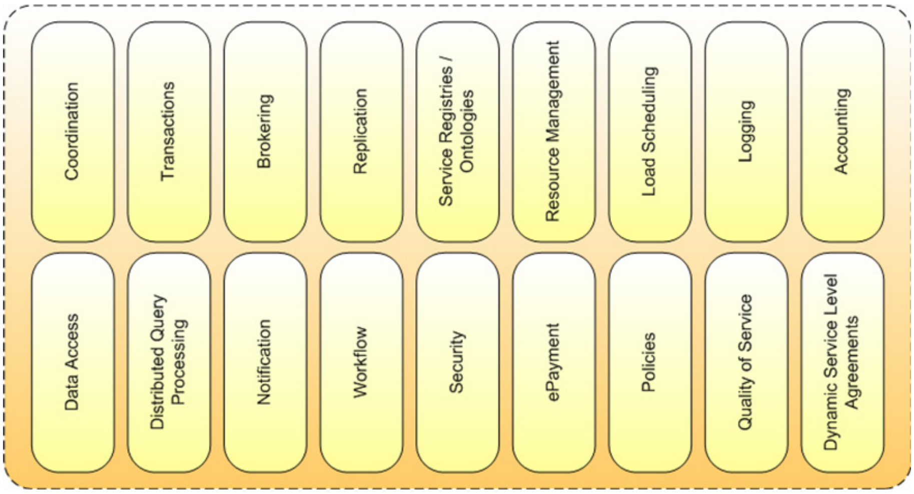

Figure 2: Provisional list of Grid Services for OGSA

which have adopted the application stack of Figure 1 and already exploit the functionality provided by the Globus Toolkit v3.0.

4. The Core Grid Middleware

The aim is to develop a software layer, the *NEReSC Core Grid Middleware* (Figure 4), on which all the e-Science application projects could be built. This will consist of a set of Grid services that will be chosen for their functionality, dependability and interoperability. It is the intention to select best-of-breed services, whatever their source. However, the current dearth of available Grid Services means that it will sometimes be necessary to take Web Services produced in the projects in which NEReSC is involved and port them to become Grid Services.

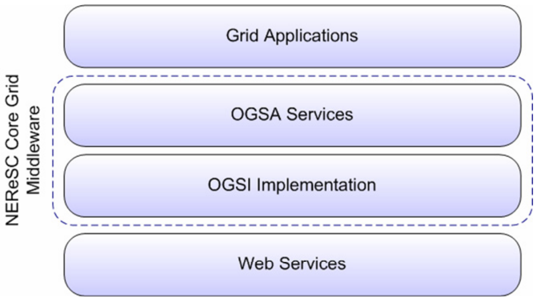

Figure 4: The NEReSC Core Grid Middleware layer

The OGSI standard is an obvious candidate on which such middleware could be based, and we chose to adopt the Globus reference implementation of the Open Grid Standard Infrastructure (OGSI) standard as the underlying platform for the Core Grid Middleware.

Even after the initial set of services has been defined, the Core Grid Middleware will not be frozen: the aim is to add new, generically useful services as they became available. The development effort would closely follow the standards work on OGSA but will not be restricted by it – particularly, in the early stages, it will sometimes be necessary to pre-empt standardisation in order to provide required services. However, the aim must be for the Grid Services implemented by the NEReSC middleware to evolve to become OGSA-compliant when the OGSA specification is finalised.

4.1. Initial Release

The initial release of the NEReSC Core Grid Middleware will be available in the end of September 2003. It will consist of four Grid services running on the Globus OGSI reference implementation. These services, shown in Figure 5, were chosen by analysis of the requirements of the NEReSC Grid projects.

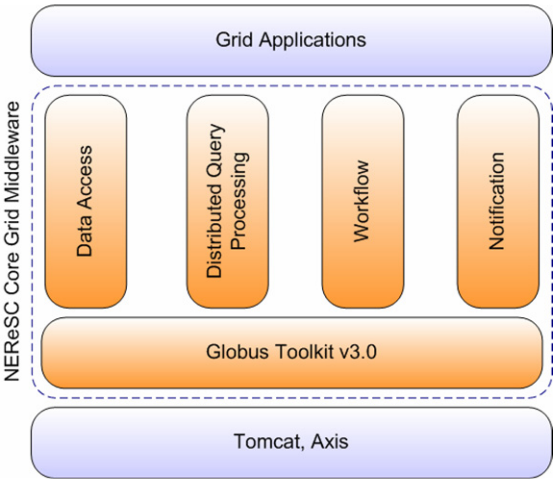

Figure 5: The initial release of the NEReSC Core Grid Middleware

They are:

- **Workflow Execution**. The ability to capture and enact computations is important for all the e-Science projects. Existing work on workflow execution [19] by IT Innovation in the ^{my}Grid project [2] is the basis for the Workflow Enactment Grid Service (WEGS) for the Core Grid Middleware. Currently, the ^{my}Grid enactment engine is based on Web Services and so NEReSC has ported WEGS to be OGSI compliant.

  The WEGS in the initial release of the Core Grid Middleware accepts workflows written in the Web Services Flow Language (WSFL) [20] and SCUFL [21]. However, an MSc project [22] has already evaluated the possibility of providing support for the Web Services Business Process Execution Language (BPEL) [23].

  NEReSC is planning to follow the work of the GGF working groups in the important area of workflow/business process composition and provide support for the Grid workflow standard when that is made available.

- **Notification Service**. The expected dynamic and distributed nature of Grid Applications means that a facility will be necessary for informing interested parties of changes in data, Grid Service status, application-specific events, etc. The Notification Service is based on that produced within the ^{my}Grid project by the University of Southampton. The ^{my}Grid Notification Service [24] provides a way for ^{my}Grid Services to publish events and/or register interest in published events. Like the Enactment Engine, the ^{my}Grid Notification Service is based on Web Services and so it needed to be adapted to Grid Service standards. The OGSI specification [4] describes a Notification portType and Service Data Elements (SDEs) through which NGS will provide the ^{my}Grid Notification Service functionality.

- **Database Access Service**. Many of the research projects in which the NEReSC is involved require access to database management systems over the Grid. The Open Grid Services Architecture – Database Access and Integration (OGSA-DAI) [10] service provides a consistent way to access relational and XML data on the Grid. The OGSA-DAI implementation is included as is in the Core Grid Middleware, since it is already built on top of Globus Toolkit v3.0 [7].
- **Distributed Query Processing Service**. NEReSC is directly involved in the design and implementation of the OGSA - Distributed Query Processing (OGSA-DQP) [12] Grid Service, which enables Grid applications to run queries on distributed data resources. A number of e-Science research projects will greatly benefit from the inclusion of OGSA- DQP in the initial release of the Core Grid Middleware.

The work required to port the ^{my}Grid Workflow Enactment and Notification Services to be Grid Services gave us valuable knowledge about the issues and effort required to do this conversion– something that we believe will be useful to the many e-Science projects that are considering carrying out this transition at some point.

4.2. Future releases

New services will be added to the initial core set over time, in response to requirements and availability. The main candidates will be the services specified by the OGSA standardisation activity, but it may necessary to pre-empt standardisation in order to meet project requirements. In most cases "best-of-breed" services will be identified and integrated, provided that an open-source implementation is available. However it may sometimes be necessary to develop services if there is a requirement and no implementation exists.

5. Web Services to Grid Services

In this section we report on experiences gained from converting the ^{my}Grid Workflow and Notification services to OGSI compliant Grid Services.

5.1. Workflow Service

The ^{my}Grid workflow enactment service offers an interface for submitting and executing a workflow. The consumer of the service is given a unique identifier for each submitted workflow. Subsequent operations, like queries about the status of the execution of a workflow, have to carry that unique identifier.

In OGSI, it is possible to create a Grid Service Instance that logically represents the execution of a workflow. Therefore a separate unique identifier is not required anymore: the Grid Service Handle (GSH) of the Grid Service Instance can now uniquely identify a submitted workflow. The submitted workflow is seen as a resource and the assigned GSH as its name. Furthermore, the status and other characteristics of the submitted workflow may be exposed through the standard OGSI mechanism of exposing the state of resources, the Service Data Elements (SDEs).

The process of mapping the design and concepts of the ^{my}Grid workflow enactment Web Service to the Core Grid Middleware Grid Service has been straightforward. However, a good understanding of the OGSI concepts was necessary in designing the Workflow Grid Service and additional, software engineering related investment had to be made on understanding and using the Globus Toolkit v3.0 (GT3).

The generic Grid Service Instance factory that comes with GT3 is used to create workflow service instances since a specialised factory was not deemed necessary.

The interface of each workflow Grid Service instance resembles that of the ^{my}Grid Web Service. It was not necessary to change the available operations since an interface for a workflow service has not yet been defined by a GGF working group. We expect that in the future such an interface would have to be standardised since a workflow service is part of the OGSA platform and an important component of many Grid applications.

5.2. Notification Service

The ^{my}Grid notification service has been designed and implemented to offer both "push" and "pull" functionality. It is based around the concept of a "topic". Interested parties are able to register to specific topics. They can either receive messages when topics are updated or they may decide to query the service for new ones.

When registering, a consumer has to identify a topic and the endpoint of the service that will be receiving the messages related to that topic from the notification service. A Grid application could consume such a Web Service without any changes but its migration to the OGSI-compliant equivalent was considered.

Two approaches were examined:

- A Notification Grid Service that just implemented the OGSI *GridService* portType but retained the same interface as the one defined by the ^{my}Grid Notification Web Service.
- A Notification Grid Service that implemented both the OGSI *GridService* and *Notification* portTypes encapsulating the functionality provided by the ^{my}Grid Notification Web Service.

The first approach would have been simpler to design and implement but it would not have provided us with any greater insight into the process of migrating to an OGSI-compliant service than the experience gained in converting the workflow service. Furthermore, the resulting service would not have adhered to the community's agreed way of doing notification, namely the *NotificationSource* portType defined by OGSI.

The Notification Grid Service utilises the Service Data Element subscription mechanism defined by OGSI. A consumer that wishes to receive notification messages must query the "topics" Service Data Element (SDE) of the Core Grid Middleware Notification Service. This SDE returns the available topics on which the service can send notifications. The information about the receiver of the messages is communicated to the notification service in the OGSI defined manner (i.e., a Locator to a service instance implementing the *NotificationSink* portType).

With the ^{my}Grid Notification Web Service it is possible to modify an existing subscription through an appropriate operation. In OGSI, the equivalent functionality is provided through a Grid Service instance which implements the *NotificationSubscription* portType.

Although the design step of the migration process is relative simple, the implementation was more difficult. The ^{my}Grid Notification Web Service is available as a servlet that due to its idiosyncrasies is more difficult to integrate with a GT3-based implementation. As a result, it was decided to create a Notification Grid Service that implemented the designed functionality by redirecting calls to the deployed, unchanged Web Service.

The software engineering work required for exposing the functionality of the ^{my}Grid notification service as an OGSI Grid Service Instance was more time-consuming than it was with the workflow service. This was due to the initial requirement of adhering to the OGSI notification interface and the idiosyncrasies of the ^{my}Grid Notification Web Service. As was the case with the Workflow Service, detailed knowledge of the OGSI concepts was absolutely necessary.

6. Conclusions and Future Work

This document has outlined the idea of the NEReSC Core Grid Middleware [1] and described the experience of designing and implementing the Workflow and Notification Grid Services by converting the existing Web Services which have been developed as part of the ^{my}Grid UK e-Science pilot research project [2].

Although it is possible to consume the ^{my}Grid Web Services from within any Grid application, it was decided that migrating them to Grid Services would provide us, and hopefullly others, with valuable experience. This migration is a process that many research projects are considering undertaking in the near future. Also, the OGSI [4] specification defines a set of characteristics and interfaces for Grid Services that it could be beneficial to utilise, like Service Data Elements, *GridService* and Notification related portTypes, a common way to name resources (i.e., GSH), etc.

The aim of this work is that by building, testing, and deploying a widely accepted set of core Grid Services, the e-Science teams would not have to spend valuable resources on developing and deploying the underlying Grid Services but could instead concentrate on building applications that utilise these services. Furthermore, valuable resources would not be wasted by different teams developing parallel implementations of the same Grid Services.

Ideally, a dedicated team of software engineers would be assigned the task of developing, hardening, testing, maintaining, and supporting the Core Grid Middleware package. To get full benefit from the package, it will also be necessary to allocate sufficient effort to all aspects of documentation, including tutorial material and courseware.

7. References

1. Parastatidis, S. and P. Watson. *The NEReSC Core Grid Middleware*. in *Proceedings of the UK e-Science All-hands Meeting 2003*. 2003. Nottingham, UK.
2. *myGrid*: http://www.mygrid.org.uk.
3. Foster, I. and D. Gannon, eds. *Open Grid Services Architecture Platform (OGSA)*. 2003, https://forge.gridforum.org/projects/ogsa-wg.
4. Tuecke, S., et al., *Open Grid Services Infrastructure (OGSI) - Version 1.0*. 2003: https://forge.gridforum.org/projects/ogsi-wg.
5. *Web Services Description Language (WSDL)*: http://www.w3.org/2002/ws/desc.
6. *The Globus Project*: http://www.globus.org.
7. *The Globus Toolkit*: http://www.globus.org/toolkit.
8. *OGSA Working Group*, https://forge.gridforum.org/projects/ogsa-wg.
9. *Global Grid Forum*: http://www.gridforum.org.
10. *Open Grid Services Architecture - Data Access and Integration (OGSA-DAI)*: http://www.ogsa-dai.org.uk.
11. *Microbase*: http://www.neresc.ac.uk/projects/Microbase.
12. *Open Grid Services Architecture - Distributed Query Processing (OGSA-DQP)*: http:/www.neresc.ac.uk/projects/OGSA-DQP.
13. *Biology of Ageing e-Science Integration and Simulation System*: http://www.neresc.ac.uk/projects/BASIS.
14. *e-Demand*: http://www.neresc.ac.uk/projects/e-Demand.
15. *GridMIST: Grid Middleware Services and Tools for Managing Resource Sharing in Virtual Organisations*: http://www.neresc.ac.uk/projects/GridMIST.
16. *GridSHED: Grid Scheduling, Hosting Environment Design*: http://www.neresc.ac.uk/projects/GridSHED.
17. *eXSys: Complex System Analysis*: http://www.neresc.ac.uk/projects/eXSys.
18. *GOLD*: http://www.neresc.ac.uk/projects/GOLD.
19. *myGrid Workflow Enactment Engine*: http://www.it-innovation.soton.ac.uk/mygrid/workflow/.
20. Leymann, F., *Web Services Flow Language (WSFL 1.0)*. 2001: http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf.
21. *Simple Conceptual Unified Flow Language (SCUFL)*: http://taverna.sourceforge.net.
22. Tamani, E., *A Workflow Grid Services*, in *School of Computing Science*. 2003, University of Newcastle upon Tyne.
23. *OASIS Web Services Business Process Execution Language*: http://www.oasis-open.org/committees/wsbpel.
24. *myGrid Notification Service*: http://fantasio.ecs.soton.ac.uk:8080/mygrid/wp2/notification/index.htm.