School of Computing Science,
University of Newcastle upon Tyne

NEReSC Core Grid Middleware

Savas Parastatidis and Paul Watson

Technical Report Series

CS-TR-815

September 2003

Published by the University of Newcastle upon Tyne,
School of Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, UK.

NEReSC Core Grid Middleware

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 Grid middleware is relatively immature and subject to frequent 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 the Newcastle projects and to design and build a common Core Grid Middleware consisting of a set of Grid Services. This paper discusses the design and development of the Core Grid Middleware package, as well as the experiences both in designing Grid services, and in porting Web to Grid services.

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 frequent 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 there is the intention to move 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 into projects.

Consequently, it was decided to analyse the requirements of all the Newcastle projects and to design and build a common *Core Grid Middleware*. The core will consist of an interoperable set of 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.

This document outlines the proposed core services and describes the process of building the Core Grid Middleware. The services that make up the core will not only be tested individually for quality but also as a set for interoperability. They will be packaged to simplify deployment and accompanied by test programs, documentation, tutorial material and courses. A testbed system running the Core Grid Middleware will be maintained and made available to all the projects, for developing and testing their applications.

Over time, further services will be added into the core. It is important to stress that it is not the intention to build all the services from scratch. In most cases, the best approach will be to identify and integrate "best-of-breed" services whatever their origin, provided that an open-source implementation is available. However, it may sometimes be necessary to develop services to meet project requirements if no implementation exists.

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.

2. Grid Services

The Open Grid Standard Architecture (OGSA) [1] 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). Currently, version 1.0 of the Open Grid Services Infrastructure (OGSI) [2], the foundation of OGSA, has been finalised.

OGSI defines the fundamental properties/characteristics of a *Grid Service* using the Web Services Description Language (WSDL) [3]. "*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*" [2]. The Globus project [4] recently released v3.0 of their Globus Toolkit [5], which, amongst other features, provides a reference implementation of the OGSI standard [2]. The toolkit also includes a number of tools for consumers and developers of Grid Services.

The OGSA working group [6] is currently considering a number of use cases [1] 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 [7] working groups have already started the standardisation process for some of these services.

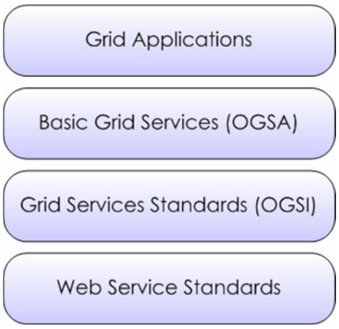

Figure 1: Grid application stack

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 [8], OGSA-Data Access and Integration (OGSA-DAI) [9], Microbase [10], OGSA Distributed Query Processing (OGSA-DQP) [11], BASIS [12], e-Demand [13], GridMIST [14], GridSHED [15], eXSys [16], GOLD [17]. 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 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.

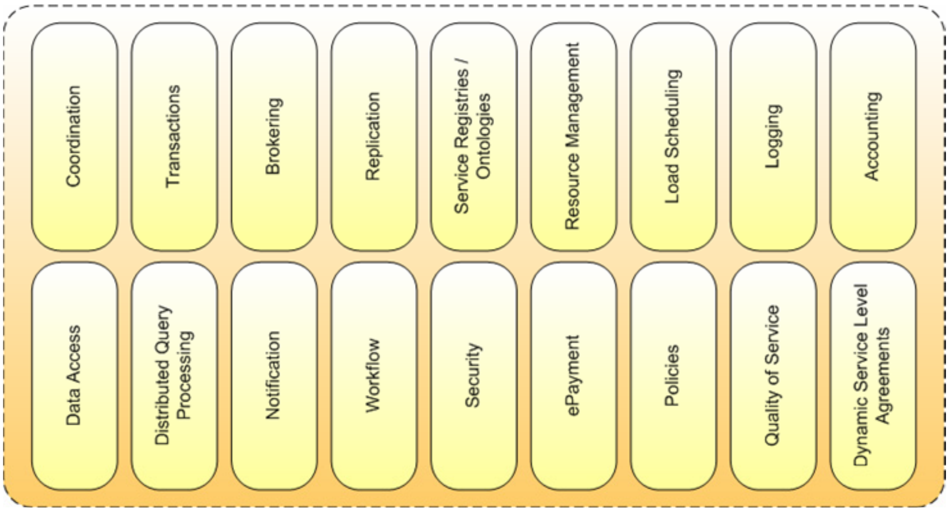

Figure 2: Provisional list of Grid Services for OGSA

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).

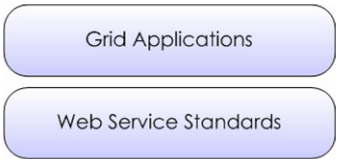

Figure 3: Application stack currently adopted by most of the NEReSC projects

Notable exceptions amongst the NEReSC research projects are OGSA-DAI and OGSA-DQP which have adopted the application stack of Figure 1 and already exploit 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.

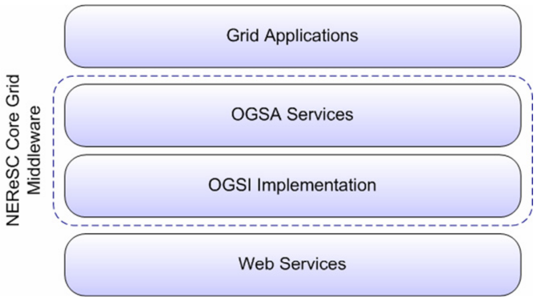

Figure 4: The NEReSC Core Grid Middleware layer

The OGSI standard is an obvious candidate on which such middleware could be based, and we recommend that the Globus reference implementation of the Open Grid Standard Infrastructure (OGSI) standard is adopted 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 preempt 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 made in 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.

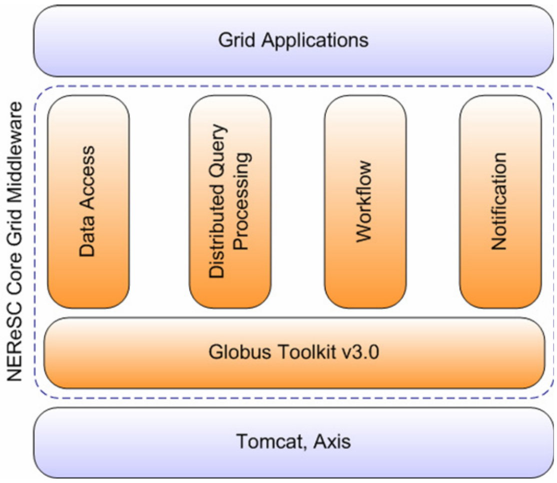

Figure 5: The initial release of the NEReSC Core Grid Middleware

They are:

- **Workflow Execution**. The ability to capture and enact computations is important for all the e-Science projects. Existing work on workflow execution [18] in the ^{my}Grid project [8] 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 is producing WEGS to be OGSI compliant.

The WEGS in the initial release of the Core Grid Middleware will accept workflows written in the Web Services Flow Language (WSFL) [19] and SCUFL [20]. However, NEReSC is already evaluating the possibility of providing support for the Web Services Business Process Execution Language (BPEL) [21].

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 will be based on that produced within the ^{my}Grid project. The ^{my}Grid Notification Service [22] 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 will also need to be adapted to Grid Service standards. The OGSI specification [2] 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) [9] 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 [5].
- **Distributed Query Processing Service**. NEReSC is directly involved in the design and implementation of the OGSA - Distributed Query Processing (OGSA-DQP) [11] 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 will give us valuable knowledge about the issues and effort required to do this conversion–something that will be useful to the many UK e-Science projects that are planning to carry out this transition at some point.

4.2. Quality Control

In order to make it as straightforward as possible for e-Science projects to utilise the Core Grid Middleware, the following activities will be carried out:

- the services will be individually tested on a set of platforms (initially Windows, Linux and Solaris);
- the set of services will be tested for interoperability;
- the Core Grid Middleware will be packaged to simplify installation;
- test programs will be provided;
- user documentation will be provided;
- tutorial material will be provided about the services;
- courses will be provided to train researchers in how to build applications on the Core Grid Middleware;
- a testbed system running the Core Grid Middleware and test programs will be made available for experimentation, and the testing of applications that build on the core services; and
- “best efforts” support will be offered to users.

4.3. 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.

In order to maintain the integrity of the core set, the activities described in Section 4.2 will need to be repeated for each new service.

5. Web Services to Grid Services

The experience gained from converting the ^{my}Grid Workflow and Notification services to OGSI compliant Grid Services is of great importance to the UK e-Science community. The Core Grid Middleware in its current state incorporates versions of the above two ^{my}Grid services with the following characteristics:

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 the submitted workflow. Subsequent operations, like queries about the status of the execution of a workflow, will 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. A unique identifier is not required anymore. The Grid Service Handle of the Grid Service Instance now uniquely identifies a submitted workflow. Furthermore, the status of a workflow that has been given to the service can be exposed through a Service Data Element.

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, additional, software engineering related investment had to be made on understanding and using the Globus Toolkit v3.0.

The interface of the workflow Grid Service resembles that of the ^{my}Grid Web Service. That was due to the lack of a GGF specification (even in draft format) on a Grid Workflow Service. It is deemed necessary that such a specification is agreed by the Grid community.

5.2. Notification Service

As with the ^{my}Grid workflow service, the design and philosophy of the ^{my}Grid notification service were mapped to the equivalent Grid Service without problems. However, since the OGSI defines a particular interface for notification services, it was deemed necessary to adhere to it. Hence, the notification Grid Service utilises the Service Data Element subscription mechanism defined by OGSI. A consumer of a Grid Service 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 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.

6. Conclusions and Future Work

This document has described a core set of grid services that will be created for the NEReSC projects. The initial release will be in September 2003, and this will be followed by updates, with other services added as required.

If there was interest outside NEReSC in the Core Grid Middleware (for example as a contribution to the ideas presented in [23] and [24]) then it would be possible to make it more widely available, but extra effort would need to be found for support. A more ambitious, but valuable, extension of the work would be to actively participate in the OGSA standardisation process and include all the Grid Services specified by OGSA in the Core Grid Middleware as they become standardised. Work would include integrating best-of-breed reference implementations into the Core Grid Middleware, but it could also extend to being proactive in developing reference implementations of OGSA Services. Further, any additional services that were not part of OGSA, but were seen as being key to the UK e-Science communities, could be developed and integrated. The advantage of this approach would be that 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 on the Core Grid Middleware. 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.

Another option for further work would be to port the Core Grid Middleware to other systems (e.g. SGI) and frameworks (e.g., .NET). There seems to be a great deal of interest in .NET as a hosting environment for Grid Services and so a .NET implementation of the Core Grid Middleware could be built. It is likely that .NET will emerge as a significant development and execution platform for Grid Services, not only on Windows but also on the Linux platform (e.g. the Mono project [25]). We are therefore closely following the progress of the two .NET implementations of OGSI [26, 27].

Finally, we believe that the Core Grid Middleware could provide valuable test-cases for the emerging UK Grid, when that moves to OGSA.

To summarise, we believe that the advantages of the Core Grid Middleware approach are:

- Easier deployment
- Centralised development
- Thorough testing
- Interoperability
- Support
- Tutorial material
- Courses and courseware
- User documentation
- Faster adoption of Grid standards

7. References

1. Foster, I. and D. Gannon, eds. *Open Grid Services Architecture Platform (OGSA)*. 2003, https://forge.gridforum.org/projects/ogsa-wg.
2. Tuecke, S., et al., *Open Grid Services Infrastructure (OGSI) - Version 1.0*. 2003: https://forge.gridforum.org/projects/ogsi-wg.
3. *Web Services Description Language (WSDL)*: http://www.w3.org/2002/ws/desc.
4. *The Globus Project*: http://www.globus.org.
5. *The Globus Toolkit*: http://www.globus.org/toolkit.
6. *OGSA Working Group*, https://forge.gridforum.org/projects/ogsa-wg.
7. *Global Grid Forum*: http://www.gridforum.org.
8. *myGrid*: http://www.mygrid.org.uk.
9. *Open Grid Services Architecture - Data Access and Integration (OGSA-DAI)*: http://www.ogsa-dai.org.uk.
10. *Microbase*: http://www.neresc.ac.uk/projects/Microbase.
11. *Open Grid Services Architecture - Distributed Query Processing (OGSA-DQP)*: http:/www.neresc.ac.uk/projects/OGSA-DQP.
12. *Biology of Ageing e-Science Integration and Simulation System*: http://www.neresc.ac.uk/projects/BASIS.
13. *e-Demand*: http://www.neresc.ac.uk/projects/e-Demand.
14. *GridMIST: Grid Middleware Services and Tools for Managing Resource Sharing in Virtual Organisations*: http://www.neresc.ac.uk/projects/GridMIST.
15. *GridSHED: Grid Scheduling, Hosting Environment Design*: http://www.neresc.ac.uk/projects/GridSHED.
16. *eXSys: Complex System Analysis*: http://www.neresc.ac.uk/projects/eXSys.
17. *GOLD*: http://www.neresc.ac.uk/projects/GOLD.

18. *myGrid Workflow Enactment Engine*: http://www.it-innovation.soton.ac.uk/mygrid/workflow/.

19. Leymann, F., *Web Services Flow Language (WSFL 1.0)*. 2001: http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf.

20. *Simple Conceptual Unified Flow Language (SCUFL)*: http://taverna.sourceforge.net.

21. *OASIS Web Services Business Process Execution Language*: http://www.oasis-open.org/committees/wsbpel.

22. *myGrid Notification Service*: http://fantasio.ecs.soton.ac.uk:8080/mygrid/wp2/notification/index.htm.

23. Fox, G., *InterGrids (Draft Concept)*. 2003.

24. Fox, G. and D. Walker, *e-Science Gap Analysis*. 2003: http://www.nesc.ac.uk/technical_papers/UKeS-2003-01.

25. *The Mono project*: http://www.go-mono.org.

26. *OGSI.NET*: http://www.cs.virginia.edu/~humphrey/GCG/ogsi.net.html.

27. *MS.NETGrid*: http://www.epcc.ed.ac.uk/~ogsanet.