Building the e-Science Grid in the UK: Advanced Grid Computing Environment

A. J. Richards, R.J. Allan, G. Drinkwater, K. Kleese

CCLRC-Daresbury Laboratory, Daresbury, Warrington, Cheshire, WA4 4AD.

Abstract

The development of an advanced Grid-computing environment has been achieved through collaborative work between the HPCPortal and DataPortal teams at CCLRC. This has been achieved using the concepts of Web services and Grid services to encapsulate core components of both portals and provide a well-defined interface for collaborative development. This work is being developed under the banner of the Integrated e-Science Environment (IeSE) project and was presented to an international audience at the recent Portals and Portlets 2003 workshop[10]

Prototype Portals

The existing HPCGrid Portal, DataPortal and InfoPortal have provided the basis for the development of the Web services and Grid services that form the advanced grid-computing environment.

Portal technology is used to allow a single point of secure access to a wide range of functionality. For DataPortal this enables searching of multiple meta-data catalogues and interaction with backend storage resource brokers (SRB). For HPCPortal the ability to search for particular types of resources or applications and interact and control job submission on remote Grid enabled machines to which the user has access. For InfoPortal to provide access to dynamic and static information about the resources currently registering with the UK e-Science Grid.

Historically the prototype portals were standalone applications, that in the case of HPCPortal and InfoPortal used a mixture of C/C++, Perl and cgi programming methods, whilst DataPortal was developed using a J2EE solution. In all cases the interface or presentation layer was directly linked to the underlying modules.

DataPortal

CCLRC's data portal, uses a multi-disciplinary metadata model of scientific data, for exploring and accessing the content of data resources within CCLRC's main laboratories in the UK and a number of other facilities in Europe. CCLRC are currently adapting the data portal for a number of e-Science projects, such as the Environment from the Molecular Level[8] so that they may store and access their own metadata and datasets and simultaneously access related metadata and datasets from other facilities around the world. To achieve this, the DataPortal was redeveloped using Web service technology so that the various services could be accessed via any user interface or system specific to the e-Science project community. Previously, access was provided only via a standard Web browser.
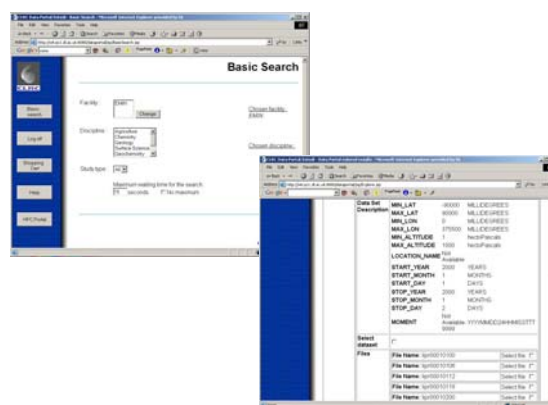

Figure 1. Views of the DataPortal web interface.

HPCPortal

HPCPortal was developed as a portal based service to give users a single point of secure access to remote resources and to simplify use

of a computational Grid. The core portal services are written using the C API of Globus Toolkit version 2. The functionality within HPCPortal allows the user to discover suitable resources via the national MDS of the UK e-Science Grid[12], submit and control jobs on remote compute resources and transfer data via a personal "sandbox" area of the portal to and from remote compute resources and remote data stores. Data links are also provided by means of direct access to DataPortal via a single sign on facility.


Figure 2. HPCPortal Web based interface showing access to the personal workspace and the ability to run remote jobs.

The prototype portal services were only available through a Web based interface. The redevelopment of the portal functionality under the banner of the IeSE project has resulted in the functionality being made available directly as Web services allowing the user to integrate portal based functionality directly within their own applications and GUIs.

InfoPortal

InfoPortal was similarly developed as a Web based user service to provide Grid resource and site-specific information. Such data would be in the form of cached data derived from the UK e-Science MDS service[9]. In addition specific functionality for querying the status of the UK

Grid via the MDS is exposed as Web services through the portal.
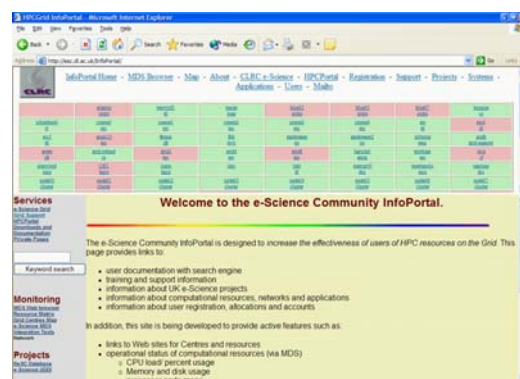
Figure 3. Web interface to InfoPortal.

Integrated e-Science Environment (IeSE)

The aim of the Integrated e-Science Environment project is to define a framework within which services can be developed that together can be used to build a portal or application.

Through the use of a Web services approach the functionality of each portal can be securely shared. By defining a layer based on these services, the underlying technology used is hidden from the user, making the use of different technologies easier to implement and enabling a "plug and play" approach to software development and usage.

The use of standard interfaces exposed via WSDL documents has allowed the DataPortal core to continue to be developed using Java and deployed using JSP technology within the Tomcat servlet container. To accommodate different requirements, including interfacing with the Globus Toolkit 2, the HPCPortal core continued to utilise Perl and C/C++ based components and the gSoap library[7] to provide the Web service wrappers.

This practice makes the portal frameworks able to integrate or migrate to newer components such as Grid services, which will be based on the use of GT3 OGSI-compliant core in a newly funded test-bed project.

At present the interaction between the portals presents the user with features such as a single

sign on, where the he or she can transparently utilise the functionality of both portals, which may be deployed on different physical servers, by presenting their credentials (invoking an x.509 certificate authentication proxy from a MyProxy server[11]) only once to one of the portals.

The ability for the portals to be able to talk to each other also allows for data selected through the DataPortal to be transferred to or by components under the control of the HPCPortal.

At a development level the visible portal interfaces can be replaced as they represent only a thin presentation layer on top of the core components of the Grid computing environment. Examples of this are the work of the e-Minerals and e-HTPX projects and CCLRC's own Integrated e-Science Environment (IeSE). In each case a customised Web based interface has been deployed that utilises particular applications from the core portal services.

**Figure 4**. Customised portal interface. Example from the e-Minerals project.

This can be further extended by allowing end users to develop their own applications and GUIs that interact with the core components of the Grid computing environment.
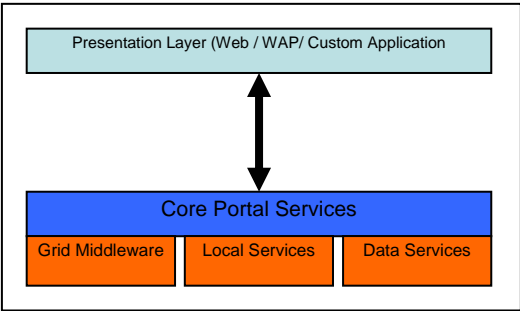
Figure 5. Schematic representation of separation of presentation layer from portal services.

The portal services are still used as a secure point of access to data and remote resources and represent a black box type of component where data output is provided in a raw format via the Web service interface so as to be formatted by the user's deployed application or interface. By adopting this design approach, flexibility is provided for customisation to suit specific project requirements but also promote code reusability. Individual components or portal services can still be developed using the language that is best suited for a particular job or matches existing requirements or requisites of a project.

Two examples of customised user interfaces separate from the core portal services are currently being developed by the Grid Technology Group. One solution uses the Jetspeed Web portal technology[5] and the concept of portlets. Each portlet contains specific functionality, essentially an application, communicating with the core portal services via the Web service interface. An example of the portlet interface for the InfoPortal is shown in Figure 6.
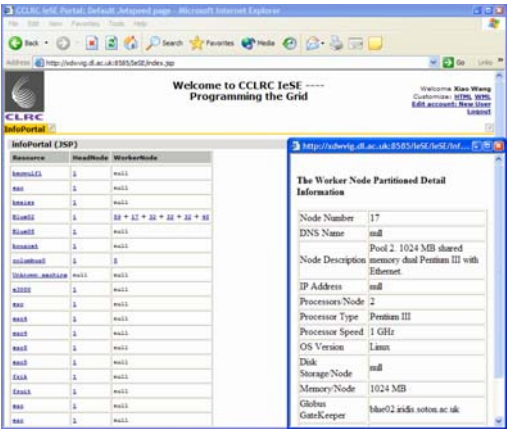
Figure 6. Jetspeed based portlet calling InfoPortal Web services (Xiao Dong Wang).

An alternative also being developed by the Grid Technology Group is the use of content management systems to act as the user portal interface. An instance of such technology is PHPNuke[6]. By default the content management system provides a modifiable modular interface that provides a range of functionality from local user management to online help systems. This has the advantage of providing the user with

"added value" services in addition to those of the core portal services. By adapting the underlying modules, the content management system has been made to work with the existing core portal services architecture for user authentication and access and control of the portal Web services. This has been achieved using PHP and SOAP to access the Web services and provides a mechanism that separates the content from the presentation layer and allows for future updates of the Web services to enable use of grid service concepts where possible.
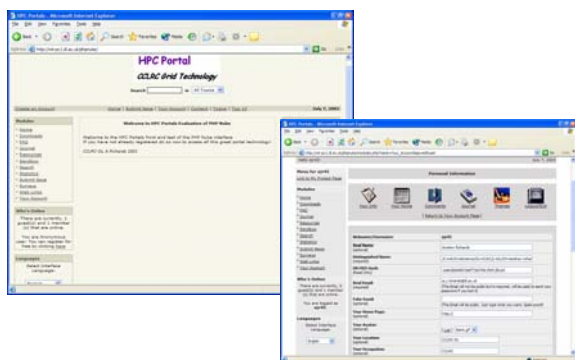
Figure 7. PHPNuke content management system acting as a portal front end.

Acknowledgements

We thank the members of the Grid Technology group and Data Management group, whose collective works form the content for the continued development of the Integrated e-Science environment.

References

1. The CCLRC e-Science centre, http://www.e-science.clrc.ac.uk/web

2. The HPCPortal reference portal, http://wk-pc1.dl.ac.uk/HPCPortal

3. The DataPortal reference portal, http://esc.dl.ac.uk:9000/index.html

4. The Globus project homepage, http://www.globus.org

5. The Jetspeed Project, http://jakarta.apache.org/jetspeed

6. The PHPNuke project, http://www.phpnuke.org

7. gSOAP http://gsoap2.sourceforge.net

8. M. Dove et al. *Environment from the Molecular Level: an e-Science Testbed Project* Proc. AHM 2003 (Nottingham 2-4/9/2003)

9. R.J. Allan, D. Chohan, X.D. Wang, M. McKeown, J. Colgrave, M. Dovey, M. Baker and S.M. Fisher *Building the e-Science Grid in the UK: Grid Information Services* Proc. AHM 2003 (Nottingham 2-4/9/2003)

10. *Portals and Portlets 2003.* International workshop held 14-17th August 2003 (e-Science Institute, Edinburgh), see http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=261

11. MyProxy certificate repository http://grid.ncsa.uiuc.edu/myproxy/

12. R.J. Allan *Building the e-Science Grid in the UK: Middleware, Applications and Tools deployed at Level 2* Proc. AHM 2003 (Nottingham 2-4/9/2003)