Portals and Portlets 2003

Rob Allan
CCLRC e-Science Centre

Chris Awre
JISC Portals Programme

Mark Baker
University of Portsmouth

Adrian Fish
University of Lancaster

e-Mail: r.j.allan@dl.ac.uk

April 14, 2004

Abstract

This report summarises current work, principally in the UK but also including examples from the rest of Europe and USA, on portals and portlets focussed on the needs of the e-Science and Information Systems communities. Many of the comments and conclusions arose during the workshop *Portals and Portlets 2003* held at the National e-Science Centre, Edinburgh, 14-17th July 2003. Examples are based on presentations at this workshop.

The agenda and presentations from the workshop, including the majority of presentation materials, are available on line at the NeSC Web sit http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=261. Links to portal projects are included in the rest of this report.

Since this meeting took place there have been a number of discussions to consider how active UK groups can best collaborate to consolidate and extend best practice and functionality of existing portals. Some suggestions are provided in the concluding section.

This report is available on line at http://www.nesc.ac.uk/technical_papers/UKeS-2004-XX.pdf

© **CCLRC 2003-4.** Neither the CCLRC e-Science Centre nor its collaborators accept any responsibility for loss or damage arising from the use of information contained in any of their reports or in any communication about their tests or investigations.

If you are reading this report and feel that your portal-related work has not been correctly represented please contact Rob Allan r.j.allan@dl.ac.uk, we will ammend the report and include your work in future surveys.

Contents

1 Introduction and Background

The Grid has not yet become a disruptive technology in its own right. However, with the use of Web-based portals for the delivery of scientific informational, experimental and computational services using a Grid infrastructure it may have the potential to be one. These are what we refer to as "active" services as opposed to the "static" services that the WorldWide Web already provides and are commonly discovered using search engines. Web services [15] have enabled limited inroads into this area, but Grid services (e.g. in an OGSA framework) promise to be a watershed. Having said that, the current proposed standards and implementations of Grid services are in a state of flux and in February 2004 at least 1 year away from being usable in "production" projects.

This section discusses the background to the international workshop held in summer 2003 to discuss these ideas and various issues concerning the deployment of portals, sharing of services and user perceptions. It focusses in particular on the impact of the JSR-168 portlet standard on portal development and the ability to share underlying tools and services.

In spring of 2003 it was felt timely to organise an international workshop focussing on portals and portlets for e-Science. This had been discussed over the previous year among members of the GGF Grid Computing Environments Research Group. With strong interest from Rob Allan and Mark Baker in the UK, Jason Novotny and Michael Russell in Germany, Massimo Cafaro in Italy and Mary Thomas, Charles Severance and Dennis Gannon in the USA, it was eventually decided to hold a 4-day workshop in the summer of 2003 hosted at the National e-Science Institute in Edinburgh, with a follow-up meeting in spring 2004 in Lecce, Italy.

The workshop did indeed prove timely for several reasons. The GridLab group had recently finished the project's new GridSphere Java portlet framework and the 4th day of the workshop acted as its first major tutorial. Discussions had started between groups in the UK e-Science and JISC portals programmes with a recognition that the two groups had much to learn from each other. Indeed Chris Awre was able to bring in a number of speakers from the JISC Information Environment community. Finally, the Java JSR-168 and WSRP standards had been proposed to W3C for ratification, which actually happened in the week following the workshop, see Section 2.2.

There have been major changes in the UK e-Science programme since the workshop and even closer links have formed with JISC. The easy delivery of access to both Grid resources and information services to end users involved in multi-disciplinary research and training is more important than ever. The concluding section in this report suggests some future actions. Some comments on technology and more recent examples cited in this report are based on information gathered for a technology a survey by Adrian Fish [20] as part of the JISC-funded ReDReSS portal project `http://redress.lancs.ac.uk`.

1.1 The Big Picture

Early portals were very application-specific, stovepipe solutions with nothing re-usable. Examples include the work of Larry Smarr, who coined the term "Grid" in 1997 and started a portals project at NCSA. The Information Power Grid portal was started in 1998 at NASA. In 1999 Mary Thomas had a group developing portals at SDSC for the NPACI Grid, IPG was still making progress and

GPDK, the Grid Portal Development Kit was started at NCSA. These were components of the US Grid Portals Project. There were also Geoffrey Fox's DOD portal and the NCSA Bioportal.

Dennis Gannon presented the Big Picture developed within the GGF Grid Computing Environment research group with particular reference to work at University of Indiana. Other material describing portal activities of the GCE-RG can be found from the GGF Web sites [1, 2]. The portal software stack was shown as four distinct layers:

1. Grid Portal – with the ability to launch, configure and control remote Grid application instances, possibly via a factory service;

2. OGSA layer – generic services such as:
   - registration and name binding
   - policy
   - security
   - data management
   - reservation and scheduling
   - event and messaging
   - administration and monitoring
   - logging
   - service orchestration
   - accounting

3. OGSI (or other) layer to link to infrastructure;

4. Resource layer with link to remote compute and data servers.

Figure 1 shows a simplified picture of a fairly typical architecture for Grid-based tools. Whilst some of the stages may be merged (typically stages 2 and 3), logically the architecture comprises:

Client: client tools and user interface. The latter may be a Web browser, GUI, drag 'n' drop environment, script or programming library toolkit. Typically there will be a firewall between the client and the front-end server so CGI and/ or web services will be used through port 80 or port 8080 via a Web cache;

Front-end server: some services may be configured on other ports which could complicate the picture. It may also be desireable to have an internal firewall to protect "private" services, e.g. access to the real databases and computational resources. For this reason a front-end server acts as a gateway. If Web services are being used, it may validate the requests and it may also implement some form of primary access control;

Back-end server: the "real" services are implemented on the back-end server which is only sent requests following the primary access control and filtering. The back-end server may implement some simple functionality for the users and also carry out important operations such as session and state management. If no additional firewall is required front and back end may be merged using a simple and fast interface. Some development tools such as WebSphere facilitate this approach;

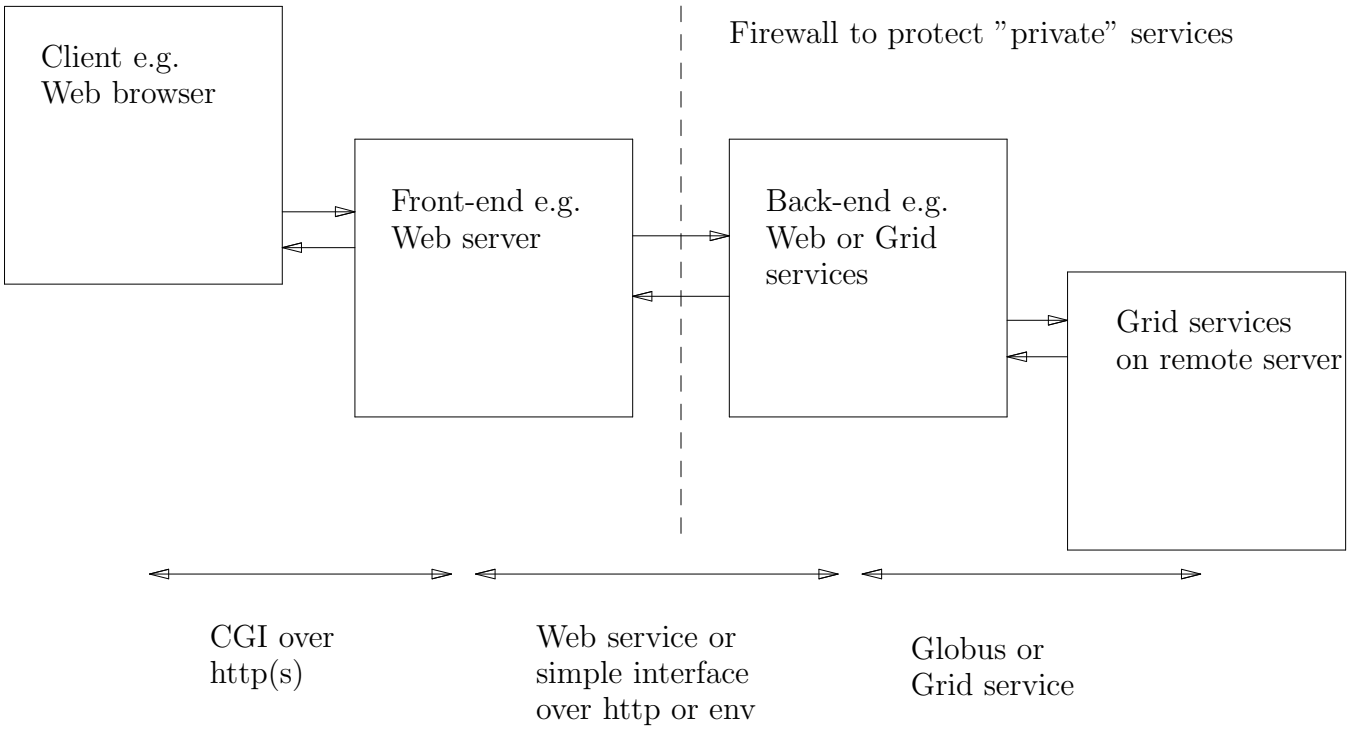

Figure 1: Four-stage Tool Architecure

Remote resource: accessed from a back-end service by a remote procedure call using Globus or other Grid middleware such as LSF, Condor or UNICORE linked into the back-end services or a Grid service interface. They are not typically exposed directly to the end user. However, for anonymous services (e.g. registry lookup) a simple Web service will suffice.

This architecture has been found to be effective in practice and permits trade-offs between security, configurability and performance.

People are currently defining what types of OGSA services are needed [22, 21]. The Grid could be defined as a collection of distributed services and a portal as a conduit to these. Solutions are built on components, where a component is a thing defined by (1) a public interface (2) semantics and a "standard" behaviour. Using a portlet framework's user interface, each component/ service can have its own interface, a portlet to uderlying services. These are "plugged" or "tiled" in a customisable fashion onto one or more views in the portal.

This picture is typical of a family of Grid portals now providing rapid development and extsibility features. A user interacts by logging on and implicitly creating a "context" or "session" which comprises his/ her recently used objects, files, jobs etc. These are represented by a set of tools for remote access and Grid services each associated with a unique portlet. Users can select the portlets they require for a particular job and to customise their portal workspace.

The Big Picture from the JISC Common Information Environment (IE) portal activities was pre-

sented by Chris Awre. For other related information see http://www.jisc.ac.uk/index.cfm?name=programme_portals, http://www.jisc.ac.uk/ie/ and http://www.jisc.ac.uk/index.cfm?name=strat_ieds0105_draft2.

The IE programme wants a small number of sophisticated interfaces for increased ease of access and use of JISC resources. The IE technical architecture was developed by UKOLN. It encompasses: content providers (institutional or commercial); fusion layer (brokers, aggregators, catalogues, indices); authentication/ authorisation (currently Athens); service registry, preference services, metadata schema registries, resolvers, institutional preferences, terminology services.

A useful description of the JISC Information Environment architecture can be found on-line at http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/. Portals are just a small part of this impressive programme of work.

There is a strategy for seamless and integrated access to digital collections. This includes connections to museums, archives and e-Science for learning and teaching. There are distinct building blocks with open standards.

The development programmes include:

- Content submission and disclosure
- Shared services
- Portals and fusion
- Presentation
- Service providers

For the JISC Information Environment portals provide the following functionality:

• Discovery tools, access search functions on (commercial) providers
• Lightweight, source remote content
• Common means of access via open standards, OAI, Z39.50
• Presentation of information
• User access via different routes: dedicated Web site, embedded functionality in known Web env
• Could be a PDA in future

We note that in this sense Google, which is probably now the most widely used “killer” search-engine application on the Web, is not a portal, but a ”gateway”, it only tells us where the information is, it does not aggregate it, although its specialised functionality, such as image searches, is growing rapidly. More information on these and related ideas is provided in an FAQ at http://www.jisc.ac.uk/index.cfm?name=ie_portalsfaq.

Some existing portal projects were presented at the workshop and are summarised below. Among other new projects are a library portal activity, including case studies and institutional portals which are being discussed for a number of Universities. There is also ongoing discussion about how to embed portal functionality in other environments e.g. via portlets. How much of the environment

should be hosted on the local resource and how much distributed? Other related work in the IE programme includes research into the disclosure of hidden content: OAI (theses, e-prints), X4L (re-purposing).

An example of a generic portal framework which was created in the Subject Portals Project for the JISC Information Environment, was described in the talk by Francisco Pinto `http://www.nesc.ac.uk/talks/261/Tuesday/PF+S=FF.ppt`. The SPP portal has the following typical functionality:

- Infrastructure for services
  - supporting software to provide access to the content
  - independent of the user's and content software
- User access
  - anytime: 24x7x365
  - anywhere: via preferred user's Web browser
  - seamless access to resources
- Built-in functionality
  - security – access management (authentication and authorisation)
  - personalisation – via user/ group profiles
  - presentation – via a variant of the Model View Control (MVC) paradigm (e.g. JSP, XSP, Velocity, XML/XSLT)

Some generic Portal engines are listed in Appendix A, Table A.

1.2 When is a Portal not a Portal?

A Grid user interface not only presents data, e.g. via generic data sources, Web cam or some kind of status monitor, but permits interaction. Authentication and authorisation mechanisms are also very important as part of an overall security model. Only user interfaces are considered therefore which permit these modes. Subsidiary modes, such as performance (e.g. for data transfer, distributed computation or visualisation), ease of use, flexibility and customisation will be considered where appropriate but may influence the actual choice of interface.

Web

Web interfaces are commonplace and therefore will be familiar to all facility users, indeed they may have found out about the facility or registered to use it via an institutional Web site (or portal). *Google* and other search engines are ubiquitous as tools for obtaining information present on the Web, and have effectively replaced encyclopaedia and libraries with a Web-based interface to distributed and constantly evolving information source.

Web sites depend on the fact that the user has a browser (Mozilla, Netscape, Konqueror, Opera, or Internet Explorer) but carry out most functionality on a server. The browser "speaks" HTTP and

will render HTML sent to it. Several additions to the basic protocol allow for instance cookies to be used for persistent communication, digital certificates to be used for authentication and JavaScript to give some necessary client-side functionality.

Web sites typically provide access to a variety of permanently on-line data linked using HTML references. CGI, the Common Gateway Interface, provides a mechanism to invoke a script or executable program running on a Web server. User input can be provided from a form, radio button, active map or similar "clickable" panel. Use of server-side CGI, JSP or PHP code can also enable dynamic pages of information but client-side JavaScript is needed to refresh the display and endow other client-side behaviour in some cases. Plugin technology exists to render known "mime-types" in special ways, e.g. to show PDF documents, display VRML or other images, show video or play audio tracks.

There are a lot of tools around to help create Web sites, e.g. Dreamweaver, or to create dynamic content, e.g. Macromedia Flash.

Portals

Portals build on the same technology used for Web sites, but enhance the functionality and flexibility to cater for the demands of specific classes of user. An interesting recent article `http://www.ariadne.ac.uk/issue37/miller/` delves more into the differences between a Web site and a portal with some examples relevant to the JISC Information Environment. A useful definition from Andy Powell of UKOLN is: "[a portal is] an online facility that provides a personalised, single point of access to resources that support the end user in one or more tasks (resource discovery, learning, research, buying plane tickets, booking hotel rooms, etc.). The resources made available via a portal are typically brought together from more than one source.
"In the context of the JISC Information Environment (IE), portals typically focus on supporting the end user in their learning and/ or research activities by providing *personalised* discovery services across multiple, heterogeneous content providers."

This view is complemented by Chris Awre, the JISC Portal Programme manager: "Technically, a portal is a network service that brings together content from diverse distributed resources using technologies such as cross searching, harvesting, and alerting, and collates this into an amalgamated form for presentation to the user. This presentation is usually via a Web browser, though other means are also possible. For users, a portal is a, possibly personalised, *common point of access* where searching can be carried out across one or more than one resource and the amalgamated results viewed. Information may also be presented via other means, for example, alerting services and conference listings or links to e-prints and learning materials."

A portal is only as good as its content. Standards are used for discovery, so content must be discoverable. There is usually a shared infrastructure, typically with registry, authentication, terminology and a presentation layer for usability, interface design, information visualisation. There must also be support through a service provider to facilitate moving of data into the portal environment.

Put simply a portal is a presentation layer which aggregates, integrates, personalises and presents information, transactions and applications to the user according to their role and preferences. It provides a persistent state/ context for each user and/ or group of collaborators. This state may consist of documents, personal registries, tool configurations, calendars, or anything else related to the user's personal configuration of the their "home" at the portal.

This kind of portal has proven to be very valuable in the business world, as summarised in another recent article http://www.esj.com/features/print.asp?editorialsId=105.

A number of commercial offerings have been produced, such as Sun's ONE Portal Server and similar offerings from BEA, IBM and of course Microsoft. See Table A.

A near-future technology for portals is "portlets," or windows onto a portal. At online portals like *Yahoo!* and *Netscape*, there are several boxes/ panes with various applications and information, such as news, stocks, weather and so on. Each of these windows is a mini-portal within the portal and is displayed via a "content management" mechanism.

These portlets are made either manually, using Java and XML, or via a portlet creation wizard that the portal-server vendor provides. Either way, XML is used to connect into back-end systems to retrieve information.

The Java portlet interface standard JSR-168 was ratified in August 2003, shortly after our workshop. Sun is pushing forward with a Java portlet API and has 18 application-server vendors supporting it (for more information go to the Web site http://www.jcp.org/jsr/detail/168.jsp). Some examples of "open" portlet frameworks are Jetspeed from Apache, uPortal, LifeRay, jPortlet (an open source project) and GridSphere from the EU GridLab project. See Table A in Appendix A.

The portlet JSR-168 specification handles the presentation end of information enabling re-use of portlets in different containers. In order for containers to present their contents as services IBM (to no surprise) is taking the lead on WSRP, the Web Services for Remote Portlets standard (also ratified in August 2003), which is based on XML and Web services. WSRP will allow portals to retrieve content from other portals via their portlet containers and other data sources. The use of WSRP and JSR-168 in a typical portal architecture is shown in Figure 2. More information on WSRP can be found at http://xml.coverpages.org/ni2002-01-21-b.html.

Content management systems also enhance Web site and portal functionality, specifically allowing users with appropriate permission to upload information which will be managed automatically using a database. Examples include CHEF (Java), PHP-Nuke and PostNuke (in PHP) and Lychee (Java).

Some of the work we have done in developing a Web portal for the computational Grid is described separately [14]. For more about the underlying technology see http://husted.com/struts/links.html.

1.3 User-Portal Interaction Model

It is perhaps useful to outline the general steps a user would take in interacting with a Grid-enabled portal as this seems to be a fairly common pattern. The interface to the user is via Web forms which may in some cases have editable default values and buttons to control subsequent actions. The technologies available to generate this interface and interact with the underlying services will be explained in the next sections. Using JSR-168 it is now common for each service to have a form presented in its own portlet. Several portlets can be assembled together onto a browser panel. Frameworks such as Jetspeed allow portlets to be made visible or not depending on user preference or other criteria.

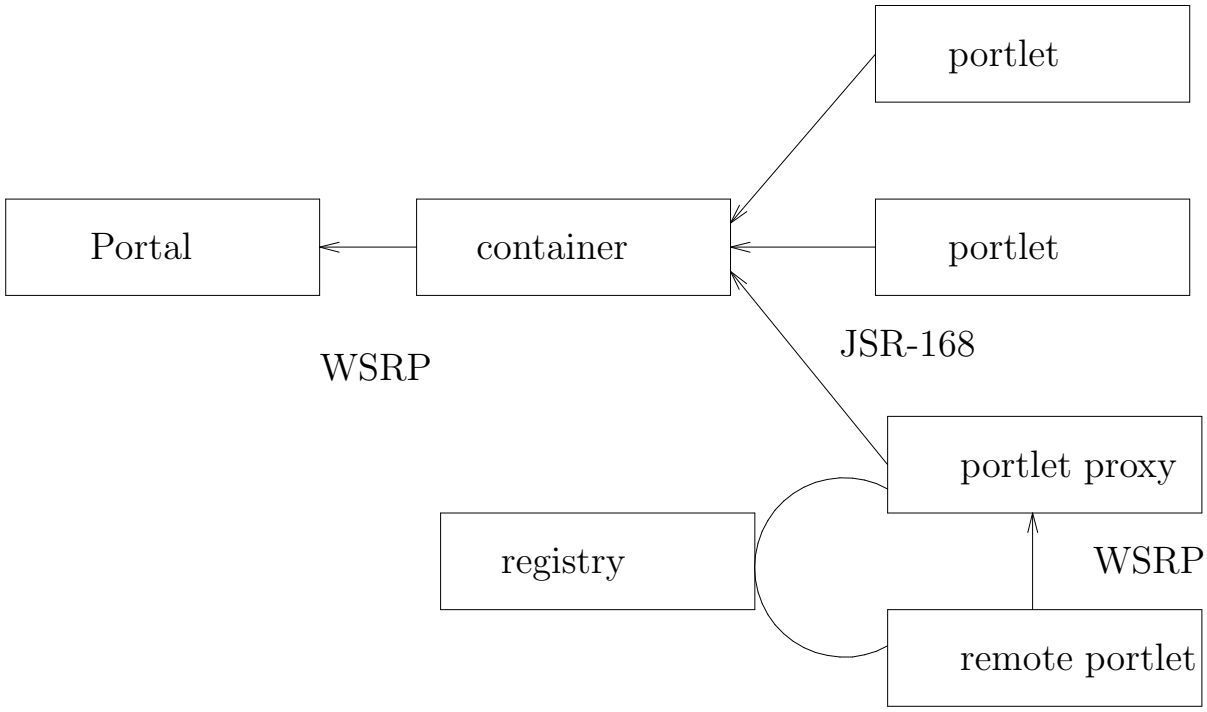

Figure 2: Relationship between WSRP and JSR-168

Firstly the user has to log on to the portal, a step often referred to as "authentication", in other words using some encrypted token or secret password as proof of identity. This step is otherwise equivalent logging into a workstation. It would bring in the user's current "context" and do other required setup in the infrastructure. Users can configure the portal to their own personal "view" and this would be stored as part of the context when they log off. One of the things to be done is to provide a "proxy certificate" to the system which can be delegated to invoke Grid services. This is almost universally done via the MyProxy manager [25].

Once logged on, the user would typically see a menu listing a range of tools (each one perhaps with its own portlet interface), e.g. as classified in Section 3.7. These tools can use an underlying content management system to share resources with other users logged onto the same portal. They can be extended using messaging systems to interface to other tools, e.g. e-mail, news feeds, etc. Portal/ project specific collaboration tools might include: calendar, thread-based discussion, on-line chat, resource space, etc. These are particularly useful for developers and are facilicated by frameworks such as PHP-Nuke, CHEF, etc. described below.

The "active services" might be made available dynamically via a Web service style API linking to middleware. In the language of Web services this would be by a query to a service registry such as UDDI (see [16]) to get the access pattern and service URI. In the language of the Open Grid Services Infrastructure (OGSI, which was being developed in 2003, now superseded by the proposed WSRF standard), it would imply that the portal code carry out introspection of a service via its GridServicePort. In either case a service "browser" could be provided, e.g. a portlet interacting with a number of Grid services via their GridSevicePorts to allow the user to select which service to use. This assumes a sufficient description is available, which in the case of a scientific application

could be rather complex. Alternatively the portal could select the latest available services when the user logs on and update its pages appropriately. Again there are many complexities if this were to be implemented in practice.

Typical steps in using the services might be: (1) upload a command file from the local workstation to the portal workspace; (2) determine a remote computational system for running an application; (3) query a database (possibly distributed) for other input data, e.g. results of a prior calculation or experiment; (4) transfer workspace contents to target; (5) transfer other data to target; (6) run chosen application; (7) check status of job later by restoring portal session; (8) transfer results back to portal workspace when finished; (9) post-process, possibly using a GUI invoked via the portal; (10) archive results along with appropriate meta-data logged in the portal system into a topic database.

Portals are not restricted to computational work, but can also provide similar interfaces to experimental procedures, possibly also involving computational analysis of data, meta-data creation and deposition in data banks.

Many of the steps link Grid-based components and other data and information resources. A powerful use of portals is to encapsulate the complex procedures described in a workflow system, typically represented as an acyclic graph. This could be used to guide even a non-expert through the decisions required to do some useful scientific work, based on steps already identified. The workflow graph could also enable more efficient scheduling of the steps to be performed (Grid optimisation). The use of workflow is discussed by Prof. David Walker in Section 2.4.

Some additional notes follow on generic aspects of these services:

The user's context:
• Keeps track of chosen applications and experiment records, tracking favourite services, notes and annotations (blogging);
• Does "session management";
• E.g. Xdirectory – a Grid context service.

Events, messages and notification:
• Message-based services are a fundamental part of the infrastructure
• Easy to transform events into directory nodes (e.g. an XML tree structure)
• Use NardaBroking and "Xmessages"
• Could interface to e-mail

How to invoke a service which has been discovered:
• Web services are designed to be invoked by machines, not users so a portal can encapsulate this (but there are easy to use human interfaces)
• WSDL describes the service (see [15]), so could use the WSDL description to create a form to interact with that Web service
• Another approach might be to use WSRP if the service has a portlet interface already.

Application "factories":
• Factory service – a persistent service that is a factory for invoking and running an application or service on behalf of the user and tracking its state
• Encapsulates complexities of platform specific environment, Grid middleware etc. • Carry out

matching of application to resource and scheduling based on user's requirements

2 More on Portal Implementation Technology

This section identifies a number of established or evolving programming technologies used for the delivery of portal services, presentation layers and other content. Web services used internally in the portals are breaking down the division between the Java and non-Java development groups. We have demonstrated inter-working of portal services coded in Java, Perl, C++, PHP and Python. This is very important as it can lead to an expansion into the scientific in addition to the business community. It will allow independent development and contribution of content and services which can be shared.

Perl, C, CGI etc.

Conventional Web sites with forms and dynamic content use CGI, the Common Gateway Interface to transmit form content over HTTP to scripts on the server. Whilst the latter are typically implemented in Perl there is a C library for CGI and the CGI environment can in fact easily be accessed and just contains a string with a series of name-value pairs separated by the & character. In addition to CGI scripts and executable programs, Web pages can be created using server side HTML. This is a good way to enforce an overall look and feel on a Web site using templates, possibly incorporating client-side JavaScript, e.g. for menus, and CSS, cascading style sheets. Several Grid portal projects, GridPort, HotPage, GRB, HPCPortal, InfoPortal have been written this way.

PHP

PHP enhances the server side dynamic capabilities. PHP is effectively interpreted code included in HTML Web pages. It is particularly useful for interfacing to databases and has therefore been used for a variety of content management systems such as PHP-Nuke, BB, DCL, RT, etc. There are however known security issues associated with careless use of PHP and CGI because it has the potential to access tho whole of the host computer's file system. See `http://www.developer.com/lang/article.php/918141`. An example of a very popular and simple forum portal for on-line discussions is phpBB `http://www.phpbb.com`.

Python

Of course for Python programmers there's Zope `http://www.zope.org`. Also take a look at ZopeZen `http://www.zopezen.org/` and Plone `http://www.plone.org`. Some popular Weblog or Blog systems appear to be written in Python, e.g. Chandler `http://osafoundation.org/technology.htm`. Chandler is branded as a Personal Information Management system.

Java, JSP, Servlets etc.

Java Server Pages (JSP) is the Java equivalent of PHP. Servlets are precompiled classes that are analogous to CGI scripts. They use however threads instead of processes to do the work, so are potentially faster (less start up time for threads than processes). JSPs are mixtures of HTML markup and Java code that are compiled on the fly by the container.

You do have to have Tomcat or another servlet container to run servlets or JSPs. It is often the case that you use Apache to serve static stuff and delegate JSPs and servlet calls to Tomcat. Tomcat contains a JSP compiler called Jasper. The result after compilation is an HTML page and servlet class combination. Containers such as Tomcat avoid the security problems associated with Perl and PHP.

Struts

Apache Struts is another Web application development framework with components for form parsing and validation etc. It pre-dates portlets (see Section 2.2). It does however give the capability of defining content including interfaces to services which can be "tiled" onto a Web page. An example of the use of Struts is the White Rose Grid portal developed by Martin Thompson of Leeds University.

2.1 Portal Services and Architecture

Generic services which could be included in a portal are typically much higher level than atomic Web or Grid services. Examples might include:

- session management
- logon/ authentication
- policy-based authorisation
- shopping cart
- query and result
- job list and status
- query-based resource discovery
- query-based application selection
- problem specification
- job composition
- service registry lookup service
- service deployment
- lifecycle management
- workflow selection and enactment
- resource scheduling
- job submission, execution, monitoring, profilling, interaction
- event notification services
- video/ audio delivery and portal-based collaborative services.

How these services is implemented is beyond the scope of this paper, but some examples were presented at the workshop.

Figure 3 is a much simplified schematic of this picture.

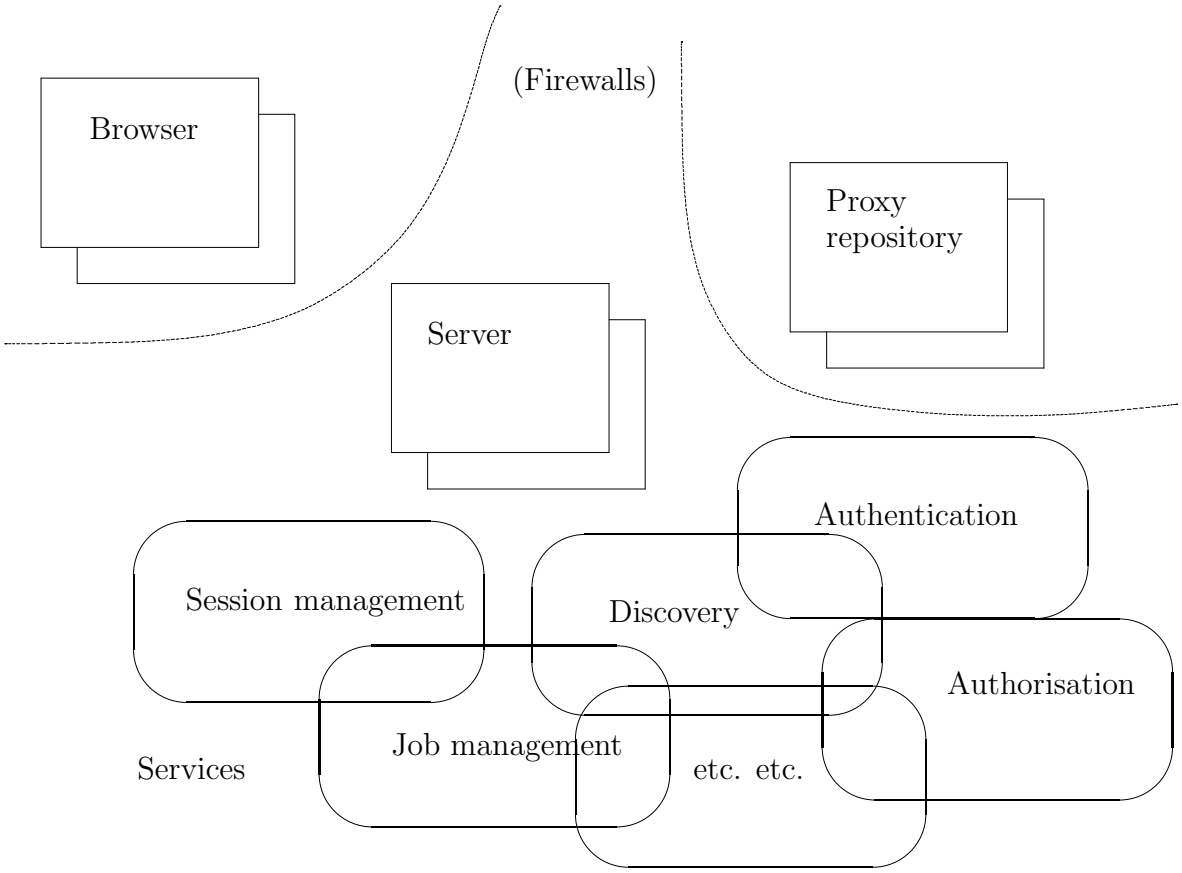

Figure 3: Service-based Portal Architecture

2.2 Portlets and Portal Frameworks

In this section we identify several frameworks for developing portals, the most promising of which are the "portlet" frameworks for which a couple of standards, JSR-168 and WSRP were ratified shortly after the July 2003 workshop.

The advantages of a portlet-based architecture are that, principally each underlying function or service can be associated with a unique portlet. This makes it easy to add new services, and many different groups can then independently contribute portlets which can be plugged into the portal. Using WSRP they can be distributed and managed remotely on many servers and the portals composed from WSDL-like information. Each user can select and configure the portlets he/ she wishes to use and selection can become part of a persistent "context". For instance certain portlets may only be useful for expert or administrative users and can be discareder by others or have their access controlled via a role-based mechanism.

The advantages of this approach were outlined by Francisco Pinto `http://www.nesc.ac.uk/talks/261/Tuesday/PF+S=FF.ppt`:

- Services

  - Web components – containers of functionality (e.g. servlets)
  - abstractions or metaphors – channels, portlets, modules, etc.

- Plugable functionality
  - common integration layer – between the content and the presentation
  - standards to access the content – Z39.50 and SRW, SOAP, XML/ RSS, SMTP/ POP3/ IMAP/ etc.
  - Standard-based – JSR-168, WSRP, etc. This ensures interoperability across different portal frameworks. It enables third party development and potentially reduces time and cost becoming crucial for an institution's core business infrastructure.

Some critera for choosing a portal framework might be:

- Integration with existing functionality
- Easy to develop new functionality
- Programming language independence
- Standards to access content
- Standards for interoperability and portability

However all frameworks are currently incomplete or have deficiencies. The standards which have now been adopted will hopefully improve the situation.

JSR-168

The Portlet Java Specification Request JSR-168 lays the foundation for a new open standard for Web portal development frameworks. Portlets define an API for building atomic, composable visual interfaces to Web content or service providers. A portlet provides a "mini-window" which can be placed within a portal page. Multiple portlets can be composed in a single page by the developer or user through the framework. Portlets extend servlets, the idea being to reuse common method signatures.

The Java portlet API JSR-168 emerged from the Java Community Process (JCP) principally from the Apache JetSpeed portal project in April 2001. JCP is an open process involving the organisation of Java developer institutions with the remit to develop and revise specifications and reference implementations for the Java platform. JSR-168 seeks to provide a portlet abstraction together with a portlet API thus enabling inter-operability between portals and portlets.

Following a public review in June 2003 JSR-168 was adopted as a full standard v1.0 in the third week of July 2003, just after the NeSC workshop.

More technical details about JSR-168 were presented by Francisto Pinto (Oxford) `http://www.nesc.ac.uk/talks/261/Tuesday/PF+S=FF.ppt`. See also `http://www.jcp.org/en/jsr/detail?id=168`.

Even in July 2003 implementations existed which were close to the standard, such as Jakarta Jetspeed, IBM WebSphere v4.2, Oracle i9AS Portal, BEA WebLogic Portal v7.0 and GridSphere. The latter, and examples of using the portlet API, were described further by Jason Novotny, Mike Russell and Oliver Wehrens `http://www.nesc.ac.uk/talks/261/Tuesday/GridSphere_Architecture_p&p.ppt` and the tutorial on GridSphere which they gave on 17th July.

WSRP

WSRP, the Web Services for Remote Portlets API defines a standard for interactive, user-facing Web servics that plug and play with portals.

WSRP emerged from the world of Web services which uses WSDL to publish service information after it was taken by an OASIS technical committee (which also reviewed the proposed JSR-168 standard). OASIS is the *Organization for the Advancement of Structured Information Standards*, a world-wide consortium that drives the development, convergence and adoption of e-Business standards. WSRP was combined with input from the proposed Web Service for Interactive Applications before a final specification was agreed in late 2002. Following a public review in May 2003, WSRP was also adopted as a full OASIS standard in the third week of July 2003.

WSRP seeks to establish a portlet abstraction with a WSDL description for how to publish, find and bind to remote WSRP-compliant services with metadata about related things such as security mechanisms, billing, etc. It is now a platform-independent bridge leveraging the language-independence of Web services and interfacing to the Java portlet API JSR-168, C# .NET API, and other WSRP implementations on J2EE or .NET. If a portlet is written to the portlet API it should be possible to publish it via a container as a WSRP service to a UDDI registry and import it into another portal using a portlet proxy.

It is important to note that you don't need a portlet server to serve WSRP compliant content. It can be served as any Web service would be (e.g. using Apache with C gSOAP, Perl SOAP::Lite etc.). This avoids content providers having to tackle issues of installing additional software like Tomcat.

Additional technical details about WSRP were presented by Matthew Dovey (Oxford) `http://www.nesc.ac.uk/talks/261/Tuesday/wsrp.ppt` and Francisto Pinto (Oxford) `http://www.nesc.ac.uk/talks/261/Tuesday/PF+S=FF.ppt`. See also `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp`. There is a white paper on WSRP available from OASIS `http://www.oasis-open.org/commitees/wsrp/wsrp_wp_09_22_2002.pdf`.

JSR-168 vs. WSRP

JSR-168 and WSRP work at different levels. JSR-168 specifies the interfaces for local portlets into their container (e.g. Jetspeed) whilst WSRP specifies the interfaces for accessing portlets across portal frameworks, i.e. from remote containers. These have to be aligned using the same notion of the objects and ability to instantiate portlets locally and remotely. Details of the portlet API have to be exposed via WSRP in order to do this. The use of WSRP and JSR-168 in a typical portal architecture is shown in Figure 2.

GridSphere

The GridSphere project is building on experience in the Java-based ASC and GPDK portal toolkits.

GridSphere provides a "white-box" framework (you edit the code) in which users can override base classes and "hook" in their own methods. It therefore requires users to become familiar with core framework interfaces which are however based on the community standard API JSR-168. Heavy use is made of design patterns which provide template solutions to commonly recurring software design problems. They also provides a common language that makes the code easier to read and understand. The Model View Control (MVC) pardigm is used to separate logic from presentation as in other portlet frameworks.

Features of GridSphere include:

• Portlet API implementation nearly fully compatible with IBM's WebSphere 4.2.
• Support for the easy development and integration of "third-party portlets"
• Higher-level model for building complex portlets using visual beans and the GridSphere User Interface (UI) tag library.
• Flexible XML based portal presentation description can be easily modified to create customized portal layouts.
• Built-in support for Role Based Access Control (RBAC). Enables managing of access for guests, users, admins and super users.
• Sophisticated portlet service model that allows for creation of "user services", where service methods can be limited according to user rights.
• Persistence of data provided using Castor JDO from ExoLab for RDMS database support, SQL and OQL
• Integrated Junit and Cactus unit tests for complete server side testing of portlet services including the generation of test reports.
• Documentation uses DocBook for HTML and PDF output of guides and tutorials
• GridSphere core portlets offer base functionality including login, logout, user and access control management.
• Localisation support in the Portlet API implementation and GridSphere core portlets support English, German, Czech, Polish, Hungarian and Greek.
• Open-source and 100% free!

Both the portlet definitions and the portal layout are coded in XML e.g. in Portlet.xml and Layout.xml and validated against schema. A portlet also has an associated class files and JSP pages. Authorised users can deploy portlets dynamically providing Tomcat >v4.1.18 is used.

The core and basic services provided in GridSphere are:

- Portlet Manager Service – Provides lifecycle methods to allow portlets to be installed, removed, initialized and destroyed by authorized users
- Login Service – Allows a User to be retrieved from a username and password
- User Manager Service – Add/Remove User Accounts; Edit User Profiles
- Access Control Service – Add/Remove User Groups; Add/Remove User Roles
- Credential Manager Service – Add/Remove allowed User Credentials; Configure use of Credential Retrieval Service

- Job Manager Service – For listing, starting, migrating, stopping jobs.
- Job Monitoring Service – Specify what to monitor for any given job and archive related information.
- File Transfer Service – For managing and scheduling file transfers.
- Data Manager Service – Access to data replica catalogues; Describe data with meta-data.
- Notification Service – Define events to be notified about; Specify how to be notified about those events.

A number of core portlets are provided by which these services can be accessed. Future plans include the provision of a complete general JSR-168 compliant framework, integration of the GAT toolkit and OGSA services, an IDE, inclusion of Flash presentations and some forms of collaboration tools.

See talk by Jason Novotny, Mike Russell and Oliver Wehrens `http://www.nesc.ac.uk/talks/261/Tuesday/GridSphere_Architecture_p&p.ppt`.

JetSpeed

JetSpeed is the open source portal framework maintained by the Apache Jakarta software group and released under the Apache Software License. It is written entirely in Java and uses XML based configuration files. Jetspeed will allow you to aggregate HTML pages, Java Applets, RSS (Really Simple Syndication) feeds, Java Servlets or JSPs and others. For historical reasons, Jetspeed already adheres closely to the JSR-168 standard.

Jetspeed is currently the framework in use in the JISC funded Subject Portals Project `http://www.portal.ac.uk/spp` and in the CHEF project at the University of Michigan `http://www.chefproject.org`. Other American Grid portal projects using Jetspeed include the Indiana University Alliance portal, the DoD PET portal, the SciDAC DOE Fusion portal and the OGCE portal.

See `http://jakarta.apache.org/jetspeed/site/index.html`. There is a JetSpeed portlet tutorial at `http://www.bluesunrise.com/jetspeed-docs/JetspeedTutorial.htm`.

jPortal

The open-source developer of this product was invited to participate in the workshop, but was unable to do so. Information on jPortal can be found at `http://www.opensource.com/jportal`.

LifeRay

LifeRay is another Java based portal framework with an impressive (for an open source product) documentation base. It has support for JSR-168 compliant portlets whereas none of the other open source portal frameworks listed in this document fully support JSR-168 – although most indicate that they will in the near future. See htmllinkhttp://www.liferay.com/products/index.jsp.

uPortal

uPortal is an open source effort using Java, XML (eXtensible Markup Language), JSP (Java Server Pages) and J2EE (Java 2 Enterprise Edition). It is a collaborative development project with the effort shared among several of the JA-SIG (Java Special Interest Group) member institutions, mainly American universities. Its development was prompted by dissatisfaction with other portal offerings available at the time.

The uPortal framework currently uses "channel" abstractions rather than portlets. Further development leading to version 3.0 will occur as part of the large US Sakai project, including addition of native support for JSR-168 compliant portlets. [1]

The uPortal channel interface however already has some similarities to the JCP portlet interface. Its channels are equivalent to "portlets" as defined in for instance JetSpeed. Remote channels, just as remote portlets using WSRP can be accesssed using a SOAP-based registration/ authentication procedure, cloned, entered, communicated with, released/ destroyed.

A channel "wizard" is built into uPortal to help install and use channels. Channels in uPortal are of pre-defined types which might include: Java applet, image, inline frame, remote channel proxy, RSS, Web proxy, XML transformation. There is also a custom type (which must be defined in a channel publishing document).

The overall look and feel of uPortal is obtained by using "skins" which can be chosen by the user and are based on the use of cascading style sheets. For remote channels to appear correctly they should use a style sheet of similar format.

uPortal is used in the commercial product from SCT `http://www.sct.com` called Luminis (as deployed in a demonstrator institutional portal study by the University of Nottingham), where it provides the basic portal functionality. SCT has also been involved in the development of uPortal, contributing code relating to the ability to push channels (portlets) to certain users at specified times. This results in a portal that can be modified by both the user and an administrator.

See `http://mis105.mis.udel.edu/ja-sig/uportal` `http://www.ja-sig.org` `http://www.jisc.ac.uk/index.cfm?name=techwatch_report_0103`.

2.3 Content Management Systems

Content management systems were not explicitly treated in the workshop, but are of considerable importance in many applications, e.g. for institutional and resources portals.

CHEF

CHEF, the CompreHensive collaborativE Framework, was developed by the group of Charles Severance at University of Michigan, USA. We are currently evaluating CHEF in the ReDReSS project, see Section 3.4.

Lychee 4

[1] uPortal currently supports JSR-168 by using the Apache project's Pluto container, and this is freely admitted to be a stopgap measure.

Lychee is a commercial Java content management system from NetCentric Europe.

PHP-Nuke

PHP-Nuke is free software available under GPL and was developed by Franciso Buzi.

PHP-Nuke is the result of many years administrating a news site called Linux Preview `http://linuxpreview.org`. First, around August 1998, Burzi wrote his own code in Perl called NUKE and used it for about 1 year. When the site got bigger, he needed a more powerful system and decided to use Slash, as used on the Slashdot site. It's good but you really need to be an expert Perl programmer to modify it, it needs many modules and uses a cpu-intensive daemon. Later Thatware was used, a good example of a news site under PHP.

PHP-Nuke is effectively a re-write of Thatware. From January 2001 to January 2002, PHP-Nuke was financially supported by MandrakeSoft, the folks that made Mandrake Linux.

Now, Burzi is still the sole developer, but receives a lot of help from the people that use and develop modules and themes via the `http://phpnuke.org` Web site. A nice example of using PHP-Nuke is the Oxford Megalith Portal `http://www.megalithic.co.uk/`. The map is particularly cool!

PostNuke

PostNuke is very similar to PHP-Nuke. See `http://www.postnuke.com`.

2.4 Workflow for Portals

Workflow is about linking components together to form a whole executable system and managing its execution. Whilst not the main focus of the workshop, this along with content management is an area of growing interest. It was felt important to see how workflow could be used to underpin portals by coordinating their services. Prof. David Walker (Welsh e-Science Centre in Cardiff) presented this section `http://www.nesc.ac.uk/talks/261/Tuesday/DavidWalkerTalk.ppt`.

Pre-defined workflows can underpin complex scientific procedures involving many components on a Grid presented to the user via a portal along with decision-making points. A key idea is that all resources are accessible as a Web or Grid service, including computation routines, access to files and databases, components of the Grid infrastructure such as the workflow enactment engines, resources monitors, etc.

A common approach to workflow is to use a visual service composition system which can create an XML description document containing information about services, interfaces and other metadata. Typically the services are registered with and discovered from a registry. Different developers can place references to their applications in this registry. Clearly one of the most important and hardest problems is to have a semantic description of applications together with their required input and output data.

The portal could be used to create as well as monitor and steer the workflow, although workflow creation might be restricted to an "export" class of users. If it is desired to give these users access to the workflow system, the visual composition tool could be presented via a portlet. A "drag and

drop" style interface would typically be used to draw the workflow using high-level applications and services as components. Data flow and control constraints are represented by the edges of the workflow graph. Special nodes may be needed to represent loops and other conditional constructs (e.g. decision points, logging) and an additional query interface, perhaps to UDDI, may be needed for service discovery. Both the latter could use portlets for user input.

A variety of XML-based languages are currently being used to describe workflow: WSFL, SWFL, SCUFL, BPEL4WS, etc. Similarly there are a number of workflow systems which could be adapted for use with portals including:

- OpenWorkflow –
- WFDK –
- Triana with VSCE, SWFL, JISGA, JACAW and GSiB – University of Cardiff
- Taverna with SCUFL – EBI and University of Southampton

More technical details and examples of workflow tools and user interfaces implemented at the University of Cardiff were presented in the rest of Prof. Walker's talk `http://www.nesc.ac.uk/talks/261/Tuesday/DavidWalkerTalk.ppt`. We note also that there was a NeSC workshop in December 2003 on *e-Science Workflow Services* `http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=261`.

3 Portal Types and Examples

This section gives some loose definitions and examples of mostly Web-based portals; the categorisation is largely derived from the workshop presentations. This attempt to classify portals is based on functional rather than technical distinctions. WSRP together with content-management systems could enable re-usable services to be developed for use in all the classes identified. As noted in the conclusions, the underlying (eventually OGSA-based) Grid services could also be presented to users in many other ways.

We do not attempt to give an exhaustive survey of portal projects, but focus on UK efforts presented at the 2003 workshop, and some more recent sigificant activities including a couple of large US initiatives.

3.1 e-Commerce Portals

Whilst not covered in the workshop, which focussed on academic applications and data collections, e-Commerce portals are very important and some are very widely used. We cannot survey this area to do it justice, so simply list a few well-known examples.

For an editorial with links to other information in this area see `http://www.esj.com/features/print.asp?editorialsId=105`.

Example e-Commerce Portals	
e-Bay	`http://www.e-bay.co.uk`
The Trainline	`http://www.thetrainline.com`
Amazon	`http://www.amazon.com`

All these implementations are proprietary, but many contain similar ideas to academic portals.

3.2 Information Resource Portals

Information Resource Portals give researchers a "view" on informational resources. These are typically large collections of records of a particular kind. Examples include: experiments or simulations performed, observations done, photographs taken, works of art recorded, census questionnaires, condition-related medical histories, etc. It is critical that the data be associated with metadata conforming to an appropriate schema which can permit meaningful searches. Cross searching multiple related collections is one important goal of these portals.

For image portals, research issues include ones related to data format diversity and others such as metadata, mapping and vocabulary. The fact that images are in diverse sizes and formats causes a problem with presentation, thumbnails are used for some purposes. Other issues relate to usability, in particular the functionality in the user interface and its applications in learning and teaching situations and for cataloguing. Other issues, not often considered in e-Science projects, are ones of rights where access control and related limitations on display and data harvesting are important. To protect the IPR in the original collections, branding and watermarking can be applied, but this may make the images less useful. Finally there may be ethical concerns of various kinds, e.g. if images are used out of their original context. Similar issues apply to other collections and for medical data privacy is of paramount importance.

Useful outputs from the JISC portal demonstrators, in addition to the stilled staff and working portals which themselves constitue a valuable resource which should be sustained, are a set of user evaluations focusing on usability issues. These can equally well guide other portal projects.

DataPortal

DataPortal from CCLRC is an example of a resource portal used in a quantitative science area. Information on this project, which uses JSP to manipulate metadata and links to distributed collections using an extensible common metadata model and XSLT mappings was given in the talk given by Rik Tyer `http://www.nesc.ac.uk/talks/261/Wednesday/DataPortal1.0.ppt`.

Go-Geo!

Go-geo! was also presented at the workshop. This was a one year JISC-funded project involving the EDINA Data library at University of Edinburgh and the UK Data Archie at University of Essex. The portal was developed to allow for the discovery of geographcal data and related resources within and beyond the HE and FE communities. The *Go-geo!* service is available at `http://www.gogeo.ac.uk/`. This is really more in the nature of a Web gateway than a portal.

PIXUS

An example of an informational portal applied to image collections is PIXUS, the JISC image portal demonstrator, which provides access to a large number of on-line digital photographs. Its portal was developed by members of the SCRAN resource team in Edinburgh and System Simulation Limited `http://www.ssl.co.uk/` to provide access to images for further and higher education, thus enriching the processes of learning and research. SCRAN manages a database of over 250 thousand digital images and has close links with the world of education and a need to provide easy to use interfaces. Whilst originally focussed on the SCRAN collection, PIXUS now integrates many other collections, including those from VADS, BioMed Image Archive, Wellcome Foundation, British Geographic Survey, and National Library of Scotland. A total of around 650 thousand images are accessable remotely via the portal.

It is important to store and use appropriate metadata about the on-line material. For PIXUS, this metadata is based on the Dublin Core, as used by the digital library community describing the key elements of many different types of data. This facilitates information discovery across a wide range of data collections through searching these common fields.

Key aspects of the PIXUS portal are its use of channels, bridges and aggregators to merge collections based on their metadata. PIXUS aggregates image collections, but does not duplicate the original material, so links back to the source for content. For users there is a "lightbox" which is an area to keep copies of the images in which they are interested.

For more information and examples from PIXUS see the talk by Sandy Buchanan `http://www.nesc.ac.uk/talks/261/Monday/1_9.ppt`. There is an on-line demonstration and evaluation of PIXUS at `http://www.scran.ac.uk/jisc/eval/`. Please try it!

SPP

The Subject Portals Project, SPP is a JISC-funded portal demonstrator project to set up a generic framework and toolset with customised pilot portals at several partner sites. The main thrust of SPP is to provide seamless access to data made available as part of the JISC RDN (Resource Discovery Network) programme. During the project, several portlets have been created, one in particular being an aggregated cross search portlet with Z39.50 capabilities for the RDN resource collections. SPP currently uses a modified version of Jakarta Jetspeed to serve the portlets, along with a proxy to convert HTTP query strings into Z39.50 search requests. It is for instance currently the portal framework used in SOSIG, the SOcial Sciences Information Gateway. SOSIG is just one of the RDN hubs.

More information about the Subject Portals Project was provided by Francisco Pinto `http://www.nesc.ac.uk/talks/261/Tuesday/PF+S=FF.ppt`.

Xgrain

Xgrain was described by Tim Stickland. It is not strictly a portal, but a cross-searching function for resource discovery which could be embedded in other portals. Xgrain can search Web resources such as those presented through other portals such as EVIL, BIOME. `http://edina.ac.uk/projects/joinup/xgrain/`.

Xgrain provides a broker which speaks SRW (the Web version of X39.50) and translates to other protocols. It has a single stable interface which could be encapsulated in a portlet. Any additional

searchable targets could be added. There is currently a GUI available for developers.

As an example, INSPEC etc. have subject-specific indices. Translation between metadata formats cannot be used to offer a cross search facility as powerful as one using the native interface. We need a Z39.50 target, WS or SRW/ CQL. Each portal needs to implement multiple channels.

Authentication is currently done via Athens, the JISC password-based scheme. Other schemes could be used via a broker passing on appropriate credentials to targets. Auhorisation is then based on existing contracts between users institutions and data providers

Control vocabularies are being investigated to narrow the search to specific subject areas, e.g. all areas that could contain a given search phrase.

SSPL

The Science and Society Picture Library is a collaboration of the National Museum of Science and Industry with the Science Museum, the National Museum of Photography Film and Television and the National Railway Museum. It provides a valuable resource gateway to the collection and links to further related social infomation. It is however a Web gateway rather than a fully functional portal.

Information Resource Portals	
DataPortal	`http://esc.dl.ac.uk/DataPortal`
Go-Geo!	`http://www.gogeo.ac.uk`
PIXUS	`http://www.scran.ac.uk/jisc/`
SPP	`http://www.portal.ac.uk/spp`
SSPL	`http://www.nmsi.ac.uk/piclib`
British Academy Portal	`http://www.britac.ac.uk/portal/`
HEIRPORT	`http://ads.ahds.ac.uk/heirport/`
Megalith Portal	`http://www.megalithic.co.uk`

3.3 Institutional Portals

These portals provide all information about an institution and the services it provides (typically educational in the case of academic institutions). This needs to be presented as both external views and to institution members (staff, students, facility users, customers, etc.). Access to the portal must therefore be via a role-based authorisation system. Roles typically include such things as: teacher, student, administrator, tutor, examiner. Institutions typically aim to build a community and attract new members/ customers using the portal as both an incentive and a single point of access to its services and information. This community might include not only work-related aspects, but leisure, as projected at universities through the local students' union activities. Various studies [3] have shown this to be useful and a number of demonstrator and prototype projects are under way.

Whilst not presented at the workshop, it is however worth taking note of the PORTAL project, Presenting natiOnal Resources To Audiences Locally. This focussed on using uPortal and developing case studies at universities of Hull, Nottingham, Bristol and De Montfort, Leicester. See `www.fair-portal/hull.ac.uk/about.html`.

Institutional Portals	
Anglia	http://evision.apu.ac.uk
Belfast	http://www.niimle.ac.uk/home.htm
Bristol	http://www.bris.ac.uk/is/projects/portal/
De Montfort	http://mle.dmu.ac.uk
Hull	http://www.digital.hull.ac.uk/
London School of Economics	http://www.lse.ac.uk/lseforyou/
Sunderland	http://my.sunderland.ac.uk/
Warwick	http://www.warwick.ac.uk/insite/

3.4 Portals for e-Learning, Awareness and Training

We sometimes consider e-Learning portals to be just a subset of the academic institutional portals, but they have a different specific aim and focus on guiding students through a structured learning experience and providing the necessary human factors support to increase the effectiveness of the portal as a means of delivery of course material. They are also concerned with testing of students' abilities as they follow the courses, so include various forms of assessment sometimes referred to as "quiz". Assignment and examination materials and results must therefore be presented in a personalised and confidential way.

Awareness and training portals are more focussed and specialise in a specific area for a more experienced audience. Self assessment and the working through customised examples then becomes more important than structured assessment. Material is provided to researchers wishing to take on board new techniques, such as e-Science, and may often teach by example rather than using formally structured material.

CHEF

CHEF is a collaboration tool aimed equally at course administrators and students. It offers administrators the facility to set up a course worksite, and students to set up study worksites. Each worksite can be configured with a number of included tools like chat, a discussion board, shared calendar and communal filespace. It makes use of the Jakarta Jetspeed portal framework to present these tools to the user in a collection of portlets. The whole thing is open source Java software, designed to run in a Web application container like Jakarta Tomcat, Jetty or IBM WebSphere and can be configured to use either file or database storage. CHEF adheres to a subset of the OKI specifications and will be integrated with uPortal v3.0 in the Sakai project. CHEF can also be clustered over a collection of server machines to improve fault tolerance.

We note that, while CHEF has gained momentum and become widely adopted in the USA as a collaboration environment for sharing course material, it has also been easily adapted to provide access to Grid services, e.g. for the NEESGrid earthquake simulation project in the USA. These two strands are referred to as "course tools" and "work tools".

Most of the CHEF development team are now part of the Sakai effort.

GENIUS

GENIUS has been used for awareness raising and is described in Section 3.7.

NIIMLE

The Queen's University of Belfast have had a working educational portal for several years. NIIMLE, a new project, was launched in February 2003.

The Northern Ireland Integrated Managed Learning Environment (NIIMLE) is a large consortium-based project, one of only two in the UK funded by the Joint Information Systems Committee (JISC). The project's main aim is to build a cross-institutional Managed Learning Environment (MLE) for the Northern Ireland region that facilitates the mobility of the lifelong learner and supports collaboration between FE and HE institutions. More specifically, the consortium's objective is to encourage students to see FE and HE in a more unified, holistic way – to look beyond the qualification for which they are enrolled and explore progression routes to higher and complementary studies, and thereafter to facilitate their continued movement within the sector. For students in the tertiary sector, the project will provide a regional portal to a range of services which students can use to inform their choices on future courses, progression routes and career paths.

Based on a brief evaluation of available technology NIIMLE is now using uPortal.

ReDReSS

ReDReSS, Resource Discovery for Researchers in e-Social Science, is a JISC-funded project for awareness and training in computational and e-Science methods. This started in November 2003 and is currently integrating content into a portlet-based framework. CHEF is being used for the prototype.

Sakai

Sakai is a development effort to create a comprehensive suite of collaboration, courseware and content management tools, based heavily on the OKI, Open Knowledge Initiative interface specifications from MIT. There is definite commitment to moving the project into the open source arena after a certain level of code stability has been achieved by the core partners. Part of the Sakai project's mission is to absorb the further development of uPortal, including making it a fully JSR-168 compliant container. The core partners are Michigan, Indiana, MIT, Stanford and the JA-SIG uPortal team. Each core partner will be contributing $2M per year over the 2 year duration of the project. There is also a contribution from the Mellon Foundation. Interested institutions can become a Sakai Educational Partner for $10K. This will entitle them to early access code releases and support services from the core Sakai development team. Sakai will not be starting from scratch, many major components from other projects have been licensed to the project. Michigan's CHEF collaboration framework (see above), MIT's Stellar course management tool, the Navigo assessment engine (see below), JA-SIG's uPortal portal framework are among them.

Charles Severance, who is chief architect of the CHEF and Sakai frameworks, was unfortunately at the last minute unable to attend the Portals and Portlets 2003 workshop. He did however visit the UK and Berlin in February 2004 to present his work and have discussions with other portal developers.

e-Learning Portals	
CHEF	http://www.chefproject.org
NIIMLE	http://www.niimle.ac.uk/home.htm
Sakai	http://www.sakaiproject.org
Awareness and Training Portals	
GENIUS	https://genius.ct.infn.it
ReDReSS	http://redress.lancs.ac.uk

3.5 Collaboration Portals

Collaboration services for portals have not really been seen in any UK portlet-based Grid project yet (some are planned in ReDReSS), but PHP-Nuke has been used in InfoPortal and HPCPortal. Prof. Geoffrey Fox of Indiana University presented exciting and novel work on portlets and Web services for collaboration and videoconferencing http://www.nesc.ac.uk/talks/261/Tuesday/Fox.ppt. Some of the work reported was carried out together with researchers at the Beihang University, China.

The approach taken in developing collaborative portals at Indiana is to build on an interoperable XML-based framework which uses Web services to create and control sessions (XGSP) and a common dynamic message environment (Narada Brokering). This enables a Global Multimedia Collaboration System to be developed (with Beihang) and a Model-View-Control (MVC) system with Jetspeed portlets. The system uses software control in distributed servers rather than hardware multicast and has been demonstrated to be efficient, scalable and flexible.

A portal for interactive collaborative working requires somewhat different services from a Grid application or resource discovery portal:

- Mechanism to set up members (people, devices) in collaborative sessions;
- Generic tools: text chat, white boards – need shared updates to text message streams;
- Audio-video conferencing – share events specifying changes in compressed streams;
- Applications: Web pages, presentations, – share key strokes to access master document;
- Visualisation – share events correspoding to changes in pixels of a frame buffer, maybe using SVG;
- Shared maps, instruments (e.g. medical).

The underlying thesis of this work is that a collaboration can be defined as a group of people sharing events which define state changes in the objects they are using. To implement this an event notification service is required and a session management service by which a user can subscribe to the original data and updates via events.

The wrapping of tools and applications as Web/ Grid services is a desirable way to share these objects. Several ways to set up and manage sessions could also be turned into services. Each service

might have a user interface which could be a portlet. Portlets could be re-directed, e.g. to PDAs, mobile phones, Access Grid or other devices.

The Global-MMCS v2.0 and XGSP MCU from Indiana University is now being developed and deployed for test purposes. Among the open-source software being used is: OpenH323, NIST SIP stack, VIC and RAT (to link to Access Grid), Narada Brokering and Java Media Framework. Jetspeed, Java Applets or ActiveX controls are used to add non-HTML clients to HTML pages. Apache Batik could be used for SVG applications. Some generic portlets, such as a VIC portlet for unicast connection to an Access Grid session could be more widely used. A Java applet could be developed to support the full multicast protocol.

More technical details and examples of the work at Indiana University were presented in the rest of Prof. Fox's talk `http://www.nesc.ac.uk/talks/261/Tuesday/Fox.ppt`. Links to additional information include: `http://grids.ucs.indiana.edu/ptliupages/projects/carousel/`, `http://www.svgarena.org`.

CHEF

CHEF was described in Section 3.4.

PHP-Nuke

PHP-Nuke was described in Section 2.3.

Collaboration Portals	
Carousel	`http://grids.ucs.indiana.edu/ptliupages/projects/carousel/`
CHEF	`http://www.chefproject.org`
PHP-Nuke	`http://phpnuke.org`

3.6 Grid Information Portals

A Grid Information System simply provides information about Grid resources (computers, databases, instruments) and their capabilities rather than using the Grid to access scientific or other information. Grid information can be divided into two kinds:

1. Static data – attributes that to not change, or only do so occasionally (e.g. after a software upgrade);
2. Dynamic data – attributes which change frequently, like current utilisation data.

Delivery mechanisms are not surprisingly different, and a Grid information portal will combine sources to create query results and present in different formats. Active maps showing resource location and attributes is a popular form.

For static data an XML or relational database could be used. snapshots of dynamic data taken over time are also regarded as static and stored in this way for later retrieval and analysis. Dynamic data is "streamed" or on demand, the Globus MDS, Monitoring and Discovery Service is an example.

Something like UDDI, used to register project services, is regarded for most purposes as static in nature.

A workshop called *Grid Information Systems 2003* was held as NeSC 24-25th April 2003 [?] `http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=190`.

GENIUS

GENIUS is described in Section 3.7. It contains a variety of monitorng and Grid information tools including active maps.

HotPage etc.

HotPage incorporates a number of different ways of collecting information about Grid resources. These include: ssh, remote commands, output to local files, Globus MDS. Problems with this includes the separation of content and presentation – some methods generate HTML, some ldif, etc. There was initially no persistent storage, just a snapshot, no MDS port and relatively poor performance. HotPage is currently using GP-IR, the XML-centric GridPort Information Repository database. It has a SOAP::Lite (Perl) Web service interface and functionality similar to R-GMA and InfoPortal. Web "scraping" has been added as a new method of data collection and live queries to GP-IR can give snapshots or retrieve historical cached info, again similar to InfoPortal.

The name *HotPage* tends sometimes to be used in a way synonymous with GridPort which is described in more detail below. HotPage actually uses the GridPort toolkit v2.3. See talks by Kurt Mueller `http://www.nesc.ac.uk/talks/261/Monday/1_4.pdf` and Tomislav Urban `http://www.nesc.ac.uk/talks/261/Tuesday/GPIREdinburghPresentation.ppt`.

InfoPortal

InfoPortal is a generic free-to-use Web-based Grid information system developed at CCLRC Daresbury Laboratory for the UK e-Science Grid [18]. It provides a variety of presentations and search facilities which can include direct links to information accessible from remote resources. InfoPortal v1.0 used standard Perl/ CGI Web technology and was closely modelled on HotPage, but used the Globus Monitoring and Discovery Service plus an XML database as sources of information about Grid resources. InfoPortal v2.0 uses PHP-Nuke as a content management framework and contains the collaborative tools that come with it. One InfoPortal module has a geographically accurate active map showing UK e-Science Centre resources.

Grid Information Portals	
GENIUS	`https://genius.ct.infn.it`
HotPage	`http://hotpage.npaci.edu`
InfoPortal	`http://esc.dl.ac.uk/InfoPortal/php`

3.7 Grid Application Portals

Grid application portals provide a variety of mechanisms to access "active" services. Some of these services will invoke remote methods to generate new data, information or knowledge through simulation, data merging or mining or download from on-line instruments possibly accompanied by

steering, visualisation and other control mechanisms. In HPCPortal, active services are classified as follows:

Class	Examples
Authentication	login, logout
Grid information	select target resources based on given search criterion
Personal workspace management	upload, download, 3rd party FTP, advance transfer, remote workspace allocation and deletion
Collaboration	calendar, discussion, chat, resouce space, mail, news, etc.
Data management	location, cache, replication, mobile code, data input, markup, mining, trigger insertion
Application management/ registry	register, discover
Job management	submission, state monitoring, logging, profiling, notification
Visualisation tools	2D, 3D, voxels, contouring, slicing, merging, application steering
Internal portal services	session, integrated state, personalisation, proxies, workflow
Other services	messaging, reservation, authorisation, accounting, billing, deployment, ontology, dictionary or controlled vocabulary

ASC

ASC, the Astrophisics Simulation Collabooratory, was developed by Mike Russell at University of Chicago, USA. The project is now led by Greg Daues at NCSA since Mike has moved with Jason Novotny to the GridLab project in Berlin. ASC, like GPDK, is entirely Java based.

GENIUS

GENIUS, Grid Enabled web eNvironment for site Independent User job Submission, is an open-source project based on the commercial (but free for academic use) EnginFrame portal platform as evaluated by the European Data Grid project. EnginFrame is also used for commercial applications, e.g. the FIAT CAE Portal. It has principally been tested for EDG by the ALICE collaboration. From 1998-1999 a graphical interface to LSF was developed for (multi) clusters for ALICE jobs. From 2000-2001 a simple but effective Web portal was developed to submit ALICE jobs to the Grid using EnginFrame and Globus. From 2002-2003 an official grant for the INFN Grid project allowed collaboration with NICE srl to integrate in a Web portal all services offered by the DataGrid middleware.

EnginFrame is built on Apache, and uses HTTPS+Java with XML+rfb (remote frame buffer) in a 4-tier architecture consisting of Web browser, EnginFrame server with Tomcat, SAP or Web services or enginFrame agents linked to services. See `http://www.nesc.ac.uk/talks/261/Tuesday/NICEGridsolution-PP2003.ppt` and `http://www.nesc.ac.uk/talks/261/Tuesday/genius_edinburgh_140703.ppt`.

GENIUS currently has over 100 services including:

- OS authentication

- File services
- Grid authentication using MyProxy
- VO selection
- Resource Broker selection
- Graphic job description
- Compute Element selection
- Parallel multi-job management
- File publication, browsing, inspection, replication, download
- Tested browsing
- P2P service using ChatEverywhere
- Grid monitoring with GridIce
- Interactive analysis GUI integration using tightVNC

A demonstration can be seen at `http://alipc1.ct.infn.it/grid/genius/gallery`. GENIUS was included in the EU EGEE proposal and is currently used as an awareness and training tool for Grid technology. In this capacity it is the official Portal of the GriDis Project for dissemination (http://web.datagrid.cnr.it/GriDis/jsp/index.jsp). Work is ongoing to include services for customisation, e.g. for PDA presentation, Grid on demand, workflow support and integration with other services. Portlet frameworks are also being investigated as an alternative platform for the GENIUS services.

GPDK

GPDK, the Grid Portal Development Kit was developed by Jason Novotny at Lawrence Berkeley National Laboratory, USA. Whilst being developed at the same time was GridPort as part of the US Portals Collaboration, it is entirely written in Java and uses servlets/ JSP. The aim was to allow developers to use GPDK components to create a customised portal for their own applications. An example of this is the portal for the UK DAME e-Science pilot project. Unfortunately Martin Thompson (University of Leeds) was unable to attend the workshop to present this.

Another GPDK-based application portal was the LCPortal developed by Mark Baker, Hong Ong and Gary Smith of University of Portsmouth.

The Liquid Crystal Portal (LCPortal)

The LCPortal is a service portal that uses a collection of Web-based tools, GridPort, MyProxy and Globus (GT2) to provide end users with a single point of access to Grid services and networked resources. LCPortal is implemented using a multi-tiered system architecture. The client browser uses a mixture of Web technologies that allows the user to manipulate and change the simulation and the visualisation parameters. The middle tier of the portal provides core services such as processing requests, dynamically generating responses, and maintaining the states of the LC computation

as well as the system. The Portal back end uses Globus services to securely access and manage distributed computation and data resources. The LC Web portal extends a users' desktop by providing seamless and secure access to remote computational resources and a simply and easy to use interface to manipulate, modify, stage, and visualise complex liquid crystal simulations. LCPortal is currently being updated to use GridSphere, portlet technologies and using a combination of Grid services based on GT3. See `http://dsg.port.ac.uk/` and [12] for further details.

GRB

GRB, the Grid Resource Broker, was written by Giovanni Aloisio, Massimo Cafaro and Ital Epicoco of University of Lecce, Italy. It uses Perl/ CGI Web technology and the underlying C API to Globus. GRB provides interfaces for a number of projects, including SAR radar imaging.

GRB is implemented as a 3-tier architecture consisting of browsers, GRB libraries and Web services linking to the Grid infrastructure via Globus.

GRB contains the following services

- User's profile management;
- Access to Monitoring and Discovery Services;
- Job Submission – interactive; batch, with support for X-windows applications, pararameter sweep, data-flow;
- Resource brokering;
- Job tracking;
- Grid Status;
- High Performance File Transfers – third-party, parallel file transfer, partial file transfer, single file and directory transfer.

At the time of the workshop the underlying GRB libraries lib_grb, lib_cookies, lib_dataflow, lib_myproxy, lib_gsift were planned for public release under GPL. Work was also planned to migrate to OGSI/ GT3 and the GridSphere portlet framework and to add support for customisation, logging and additional job scheduling and checking.

See talk by Massimo Cafaro `http://www.nesc.ac.uk/talks/261/Monday/1_7.pdf`.

GridPort

GridPort was one of the earliest Grid portals developed by Mary Thomas and Steve Mock at the San Diego Supercomputer Center, USA. It was based on traditional Web technology, coded in Perl and using CGI. It originally used SSH, and later Globus to access distributed resources. GridPort is the primary interface for many users of the NPACI Grid and has been customised for several applications, such as GAMESS-US, Telescience, BIRN, Pharma, etc. The current release of GridPort is v2.3 which is written in object-oriented Perl with modular packaging and simple installation to run on the Apache Web server.

GridPort v3.0 was planned to support OGSA services and be available as a source distribution for rollout to users of the PACI, NPACI, NCSA, and PSC resources (the Enhanced TeraGrid). Portlet technology was being added using Jetspeed by wrapping GridPort 2.3.1 services. In this case the portlets and servlets rendered by Jetspeed talk via HTTP to a CGI wrapper layer which is part of GridPort. This in turn accesses Globus GT2 services on remote systems.

See talks by Kurt Mueller `http://www.nesc.ac.uk/talks/261/Monday/1_4.pdf` and Tomislav Urban `http://www.nesc.ac.uk/talks/261/Tuesday/GPIREdinburghPresentation.ppt`.

GridLab

Developed the GridSphere framework for its portal, see Section 2.2.

HPCPortal

HPCPortal is a generic free-to-use portal platform developed at CCLRC Daresbury Laboratory. HPCPortal v1.0 used standard Perl/ CGI Web technology, but v2.0 uses PHP-Nuke as a content management framework. HPCPortal was originally closely modelled on GridPort but has now diverged and uses different technology and is based on Globus with Web service wrappers. The C API to Globus is used in a similar way to GRB. HPCPortal has been customised for the e-Minerals e-Science pilot project. See talks by Andy Richards [9, 17, **?**] `http://esc.dl.ac.uk/HPCPortal`. HPCPortal services were described above.

OGCE

In the USA several groups are now collaborating in the Open Grid Computing Environment (OGCE) Portal Project funded by the National Science Foundation (NSF) as part of its middleware initiative, indeed it is referred to as the NMI portal. This includes researchers and end users from Indiana, NCSA, Michigan, Texas and Argonne National Lab. This colloration has grown from the previous portals project (which developed the Perl-based NPACI GridPort and Java-based GPDK from NCSA). It is a response to the need to embrace the new technologies of portlets and Web/ Grid services to serve the needs of NCSA, NPACI, DOE and NASA Grid users. Jetspeed, CHEF and OGSA/ OGSI are currently being used with the Argonne Globus Java CoG kit. There is also an evaluation of GridSphere.

Existing services and portlets in OGCE are:

- MyProxy – MyProxy portlet;
- File management – GridFTP portlet;
- Resource verification;
- LDAP browser;
- Grid job launch and monitor;
- Collaboration – news, groups, mail, messaging (NaradaBroker and Xmessages) Java JMS etc.

In general OGCE (which is also sometimes called the NMI portal project), represents a union of many of the American Grid related portal projects. Dennis Gannon notes that it has a long way to

go over the next twelve months, including its integration into Sakai and the port of all the portlets to JSR-168 and futher consideration of WSRP. Currently much of the effort of the OGCE project is focused on building Grid/ Web services and their access by client interfaces. Some of the work at Indiana has the goal to deploy Grid services for things like workflow tools and make it possible for portal users to discover and load the client interfaces into their portal environment or compos it into applications as components.

Planned portal deployments that are based on OGCE include:

1. The NSF TeraGrid portal (prototype at `http://www.extreme.indiana.edu:18081/teragrid/portal`);
2. The Linked Environments for Atmospheric Discovery NSF ITR project (prototype at `http://lead.extreme.indiana.edu:10081/lead`);
3. The DOE Fusion Portal. see: `www-fp.mcs.anl.gov/middleware-review/ProjectReports/PortalWebservicesfinal.pdf`;
4. The Southern California Eathquake Center see: `http://epicenter.usc.edu/cmeportal/proposal.html` for the project which is just starting its portal effort;
5. The NEES Grid portal is based on CHEF and not OGCE but it was the portal effort that first used many components that are now going into OGCE;
6. the NCSA Alliance Portal, see: `https://portal.extreme.indiana.edu:8443/alliance/index.jsp`

The current version of OGCE is based on CHEF v1.0 and will be ported to Sakai by autumn 2004 so that it will be JSR-168 compliant. The generic OGCE release is available from `http://www.ogce.org`.

PROGRESS HPC Portal

PROGRESS is a large Polish initiative to construct a national Grid system. It has compute and data servers at several locations including Poznań, Lódź and Kraków. Because Sun Microsystems are a project partner the PROGRESS Grid uses Globus with Sun Grid Engine and the Sun One portal server v6.0.

A HPC portal was developed for PROGRESS at the Poznań Supercomputing and Networking Center, Poland. It uses the SunOne portal platform and presents a range of data and computational services to the users. SOAP and FTP are used as the main transport mechanisms to access these services and portlets with XSLT are used for presentation. The portal is implemented using four major components:

1. WP: Web portal
2. GSP: Grid Service Provider
3. GRB: Grid Resource Broker

4. DMS: Data Management System

This results in a 4-tier architecture comprising: portal; Grid service provider; Grid management system; and remote resources.

Using the GSP layer, different customised portals can be built and other interfaces provided to access the same services. Services are currently as follows:

- Job Submission – create jobs, submit to Grid, monitor;
- Application Management – provide information on Grid applications, register new applications;
- Provider Management – information on other services, e.g. news service, discussion.

These have Web service interfaces. We note that authentication and authorisation services are provided by SunOne and the DMS.

See talks by Maciej Bogdański `http://www.nesc.ac.uk/talks/261/Tuesday/progress-sunoneportal.ppt` and `http://www.nesc.ac.uk/talks/261/Tuesday/progress-edinburgh.ppt`. See also `http://progress.psnc.pl` and `http://www.iplanet.com/products/iplanet_portal`.

Grid Application Portals	
AliEn	`http://alien.cern.ch/`
ASC	`http://www.ascportal.org`
Cactus	`http://www.cactuscode.org`
DAME Portal	
GENIUS	`https://genius.ct.infn.it`
GRB	`http://sara.unile.it/grb/`
GridPort	`http://gridport.npaci.edu`
GPDK	`http://www.doesciencegrid.org/Projects/GPDK`
GRIP	`http://www.grid-interoperability.org/`
GridLab	`http://www.gridlab.org`
HPCPortal	`http://esc.dl.ac.uk/HPCPortal`
LCPortal	`http://dsg.port.ac.uk/`
NEESGrid Portal	
OGCE	`htp://www.ogce.org`

4 Conclusions and Way Forward

It is clear that there is a lot of established expertise and momentum in the UK to develop Web-based portals for a variety of purposes. We have established strong links and potential collaborations bridging the UK, USA and other European developers and now also bridging the e-Science and JISC communities. It is important to continue this work and lead identified areas which will be taken via the Global Grid Forum research and working groups as input into the definition of standards leading to software sharing.

There are a number of UK groups already actively developing re-usable resource and Grid-based portals and portlet services. There are actually a large number of other groups developing informational, institutional, e-Learning and awareness and training portals. In many cases the frameworks being developed could be shared and the underlying portlets and services could be re-used if an appropriate architecture and standards were adopted. This implies a portlet framework plus a message-based service approach rather than a methods-based approach (advice from Geoffrey Fox, 26/2/04).

Based on our experiences and outcomes of the NeSC workshop we recommend the following:

1. Portal services development should be recognised as a strength of the UK middleware initiatives, e.g. as noted by Fox and Walker their *Gap Analysis* [7];
2. The active UK groups should collaborate. These include developers from the three JISC pillars: support for research, teaching and learning and the Common Information Environment plus the e-Science and Particle Physics Grid user communities;
3. UK developers should continue to work with the GGF Grid Computing Environments research group and the American Open GCE and Sakai projects plus other relevant international fora;
4. Developers should save effort by sharing services and methodologies and customising the existing Web-based presentation layers for delivery to all end-user projects;
5. The UK should be active in defining, classifying and developing portal services for input into the OGSA space between (possibly changing) infrastructure and application layers;
6. A range of toolkits (thin clients, portals, scripting languages, GUIs etc.) should be developed to extend and simplify access to Grid resources and information systems leading to the eventual emergence of one or more interfaces to a Virtual Research Environment.

It seems that there is a good possibility of linking future activities in the areas we have described more closely with developments in other sectors, including institutional and learning and teaching arenas. There does appear to be a convergence in technology and benefits from sharing some of the tools being developed. To this end we are looking to JISC, the Joint Information Systems Committee, to fund a small number of evaluation studies to examine issues of re-usability and inter working of components in the emerging portlet frameworks.

5 Acknowledgements

We thank all the speakers at the *Portals and Portlets 2003* workshop. In particular we thank Mike Russell, Jason Novotny and Oliver Wehrens for their help in organising the workshop and for spending the extra time to give the GridSphere tutorial on 17th July which was very enjoyable. You can find their presentation material for that on the NeSC Web site `http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=261/`. We thank Dennis Gannon of University of Indiana for his help and enthusiasm both during and after the workshop.

Although unable to attend the workshop in person owing to family committments, we thank Charles Severance of University of Michigan for his support and input into this report and discussions on CHEF and Sakai during his visit to Europe in February-March 2004. We also thank Martin Thompson and Peter Dew of University of Leeds for additional input on the White Rose Grid and DAME portals and Struts. Other additional input has come from the SPP Portal project which we have discussed on several occasions with Francisco Pinto, Mike Fraser, Marieke Guy and Jasper Tredgold. Still more additional input has come from Geoffrey Fox who has been gracious to answer our e-mails on many occasions and share information with us both in private and at other meetings.

Also since the workshop Rob Crouchley of Lancaster University has become a champion for portals, especially for training and awareness in e-Social Science with his ReDReSS project. Some of the technical material surveyed in this project has been included in the separate project report by Adrian Fish.

Finally we thank all the people we have spoken to over the last few months about portals, particularly those in JISC who have had the patience to listen to our ideas. In this context we thank Tony Hey, Alison Allden and Sarah Porter for their suport.

References

[1] A report on grid computing environments which was an early GGF GCE document: *Overview of Grid Computing Environments* `http:/grids.ucs.indiana.edu/ptliupages/publications/ggf-gce-overview2.doc` which has lots of survey stuff about Grid portals from the "early days" of 2001. This is also GGF document so you can find the final pdf on the GGF documents web site at: `http://forge.gridforum.org/docman2/ViewCategory.php?group_id=90&category_id=28`

[2] *Grid Portals: A Scientist's Access Point for Grid Services* GGF Community Practice Document, GCE-RG. `http://www.extreme.indiana.edu/~gannon/ggf-portals-draft.pdf`

[3] *Hull portal survey*

[4] JISC Portals Workshop, Oxford Hotel (May 2003)

[5] ACE

[6] Subject Portals Project

[7] G. Fox and D.W. Walker *Gap Analysis*

[8] R.J. Allan et al. *Level 2 Grid – Final Report* (ETF, April 2003)

[9] R.J. Allan et al. *Middleware, Applications and Tools deployed at Level 2* Proc. AHM'2003 (Nottingham, September 2003)

[10] R.P. Tyer, L.J. Blanshard, K.K van Dam, R.J. Allan, A.J. Richards and M.T. Dove *Building the e-Minerals Mini-Grid* Proc. AHM'2003 (Nottingham, September 2003)

[11] M. Baker, H. Ong and G. Smith *A Status Report on developing a Grid-enabled Web Portal using Commodity Toolkits* (University of Portsmouth, 2001) `http://esc.dl.ac.uk/GridWG`

[12] M. Baker, H. Ong and G. Smith *A Portal for Liquid Crystal Modelling Applications* (University of Portsmouth, 2002) `http://esc.dl.ac.uk/GridWG`

[13] *HPCPortal* R.J. Allan

[14] R.J. Allan *Developing Web Portals for the UK e-Science Grid.* technical report (UKHEC Collaboration and CCLRC e-Science Centre, 2002) `http://esc.dl.ac.uk/GridWG` `http://www.ukhec.ac.uk/publications/reports/portal_guide.pdf`.

[15] Rob Allan, Dharmesh Chohan, Xiao Dong Wang and Daniel Hanlon *An Introduction to Web Services and related Technology for building an e-Science Grid* (Grid Technology Group, Daresbury Laboratory, 2002). `http://esc.dl.ac.uk/ETF/public/WebServices/index.shtml`.

[16] R.J. Allan, D. Chohan, X.D. Wang, M. McKeown, J. Colgrave and M. Dovey *UDDI and Web Services Inspection for e-Science* (UK Grid Support Centre, 2002). `http://esc.dl.ac.uk/WebServices`.

[17] A.J. Richards, R.J. Allan, G. Drinkwater and K. Kleese *Advanced Grid Computing* Proc. AHM'2003 (Nottingham, September 2003)

[18] R.J. Allan, D. Chohan, X.D. Wang, A.J. Richards, M. McKeown, J. Colgrave, M. Dovey, M. Baker and S.M. Fisher *Grid Information Services* Proc. AHM'2003 (Nottingham, September 2003)

[19] S.J. Newhouse et al. *ICENI*

[20] Adrian Fish *ReDRESS Project Review: Survey of Portals and Related Software.* (University of Lancaster, February 2004)

[21] Dave Berry et al. *UK Experience with OGSA.* Report on the workshop held at CCLRC, May 28-29th 2003. `http://www.nesc.ac.uk/technical_papers/UKeS-2003-02.pdf`

[22] Bill Johnston and John Brooke *Core Grid Functions: a Minimal Architecture for Grid.* White paper of GGF GWD-R track of Grid Protocol Architecture working group. `http://grid.lbl.gov/GPA/`

[23] D. Gannon, M. Christie, O. Chipara, L. Fang, M. Farrellee, G. Kandaswamy, W. Lie, B. Plale, A. Slominski, A. Saarangi, Y. Simmhan, *Building Grid Services for User Portals*, GGF Workshop on Grid Services, Chicago, (Oct. 8th 2003)

[24] S. Krishnan, R. Bramley, M. Govindaraju, R. Indurkar, A. Slominski, D. Gannon, J. Alameda, D. Alkaire *The XCAT Science Portal* Proceedings SC2001, Nov. 2001, Denver. Extended version in Scientific Programming 10(4) (2002) 303-317

[25] J. Novotny, V. Welch *MyProxy* NCSA

A Generic Portal Engines

Tables A and A lists a number of generic portal engines and some related software from commercial vendors and public-domain developers.

Table 1: Generic Portal Engines 1

Commercial	
ASP.NET	http://ibuyspy.com
Blackboard Community portal	http://products.blackboard.com/cp/bb5/index.cgi
Campus Pipeline Luminis	http://www.campuspipeline.com/
CA CleverPath Portal	http://www3.ca.com/Solutions/Product.asp?ID=262
CMIS ePortal	www.ccmsoftware.com/cmis.htm
Elipva	http://www.elipva.com
EnginFrame	http://www.enginframe.com
Epicentre	http://www.epicentric.com
IBM WebSphere Portal	http://www-4.ibm.com/software/webservers/portal/
WebSphere Portal API	http://www7b.software.ibm.com/wsdd/zones/portal/portlet/4.1api/
Jahia	http://www.xo3.com
Lotus Notes Domino	http://www.lotus.com/
Lychee	http://www.netcentriceurope.com/content/product_overview.htm
Macromedia Cold Fusion	http://www.macromedia.com/software/coldfusion/
MediaApps NetPortal	http:/www.mediaapps.com/
Merant Collage	http://www.merant.com/Products/WCM/collage/home.asp
Microsoft Exchange 2000	http://www.microsoft.com/exchange/default.asp
Microsoft Sharepoint	http://www.microsoft.com/sharepoint/
Mongoose Portal Studio	http://www.mongoosetech.com/products/portalstudio.html
Novell Silverstream ePortal	http://www.silverstream.com
Novell Portal Services	http://www.novell.com/products/portal/quicklook.html
Oracle 9iAS Portal	http://otn.oracle.com/products/iportal/
Oracle Portal	http://portalcenter.oracle.com
RedHat Portal Server	http://www.redhat.comn/software/rha/portalserver/
SITE e:Vision	http://www.sits.co.uk/
Sun iPlanet Portal Server	http://www.iplanet.com/products/iplanet_portal
Unicon Academus	http://www.uicon.net/academus
WebCT Vista	http://www.webct.com/products/viewpage?name=products_vista

Table 2: Generic Portal Engines 2

Public Domain	
CHEF	http://www.chefproject.org
Enhydra	http://www.enhydra.org/
ExoPlatform	http://tuan.dyndns.org/exo/faces/public/portal.jsp
FreshMeat PHP Portal	http://freshmeat.net/projects/phportal/
Gluecode Portal Foundation Server	http://www.gluecode.com/website/html/PFS.html
Grid Portal Development Kit	http://dast.nlanr.net/Projects/GridPortal/
GridPort	http://gridport.net/index.cgi
GridSphere	http://www.gridsphere.org
JetSpeed	http://jakarta.apache.org/jetspeed/site/
jPortlet	http://jportlet.sourceforge.net/
LifeRay	http://www.liferay.com/products/index.jsp
MyLibrary	http://www.lib.ncsu.edu/eresources/mylibrary/
PHP-Nuke	http://phpnuke.org
PostNuke	http://www.postnuke.com
Sakai	http://www.sakaiproject.org
Thatware	http://thatware.org
uPortal	http://mis105.mis.udel.edu/ja-sig/uportal
Zope	http://www.zope.org/