



SAKAI EVALUATION EXERCISE

(A Report to JISC).

Rob Crouchley and Adrian Fish
e-Science Centre of Excellence, University of Lancaster

Rob Allan and Dharmesh Chohan
CCLRC e-Science Centre, Daresbury Laboratory

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

December 31st, 2004

Abstract

This report summarises an evaluation of various aspects of the CHEF, Sakai and other related projects that may play a significant role in establishing a Virtual Research Environment (VRE) for the UK.

The work carried out built on e-Science activities at Lancaster and Daresbury, which include software development and hardware deployment for collaborators in the physical sciences, substantive e-Social Science research and social science training and awareness-raising. It built upon the existing collaborations that the project team have with each other and with other university groups in the UK. It is now using existing mechanisms to disseminate the outcomes of the work. Throughout the period of this evaluation we have received clear expressions of interest in working with us from the developers of Sakai (<http://www.sakaiproject.org>). This included discussions and presentations at the Sakai Developers Workshop, Denver, USA, 23-27/6/04 and subsequently at the JISC-CETIS workshop, Oxford 4-5/11/04.

The evaluation exercise was greatly facilitated by Lancaster deploying a CHEF implementation (<http://redress.lancs.ac.uk>) from day one and later the Sakai RC2 implementation to use as a collaboration tool for developers based at Lancaster and Daresbury.

This report summarises the outcomes and contains the conclusions of the evaluation which covered:

1. Comparing Sakai/ CHEF with Alternative Frameworks for VREs;
2. Establishing the ease of administration (EoA) of Sakai/ CHEF for a VRE;
3. Establishing the feasibility of making existing VRE (Grid) components available via Sakai/ CHEF;
4. Establishing the feasibility of extending the functionality of Sakai/ CHEF particularly to use Web Services for distributed development and deployment.

This report is available on line at <http://redress.lancs.ac.uk:8080/portal> under “SEE/Resources” and http://www.grid.ac.uk/Sakai/sakai_doc.pdf [PDF] or http://www.grid.ac.uk/Sakai/sakai_doc [HTML].

Executive Summary

This is a report of the JISC-funded Sakai Evaluation Exercise and is organised as follows. Firstly there are sections giving the background to portal and portlet technology and introducing the requirements and expectations for a VRE. The conclusions of our evaluation work are also explained and a way forward in using Sakai to address the needs of VRE developers is presented along with some architectural suggestions.

- Introduction – background to this report.
- Portals and Portlets 2003 – summary of the *Portals and Portlets 2003* workshop held at NeSC 14-17th July 2003.
- Sakai SEPP Developers' Conference – trip report from the first SEPP conference.
- Role of Portals in a Virtual Research Environment – background material on VREs and the Open Service Architecture.
- Sakai Evaluation including CHEF, OGCE and GridSphere – capabilities of the chosen frameworks and summary of technical evaluations.
- Additional Comments on Portal Architectures.
- Conclusions.
- Acknowledgements.

Secondly there are appendices which give more detail of the work carried out and further background information.

- WP 1 – Comparing Sakai/ CHEF with Alternative Frameworks for a VRE.
- WP 2 – Ease of Administration (EoA) of Sakai/ CHEF for a VRE.
- WP 3 – Making existing VRE (Grid) Components available in Sakai/ CHEF.
- WP 4 – Issues involved in extending the Functionality of Sakai/ CHEF.
- Integration of UK VRE Tools into Sakai.
- Resources.
- Service Capability Set.
- Generic Portal Engines.
- Another independent Technical Report.

The internal Sakai architecture is still in flux, which means that we have carried out less tool porting than originally planned, we mainly used CHEF instead. Nevertheless the user interface is stable and was liked by the groups to whom we have given demonstrations. We were able to modify both CHEF and Sakai RC1/ RC2 and Sakai v1.0 to use a PostgreSQL database and were able to port the ReDRESS portal content from CHEF into Sakai RC1/ RC2 and Sakai v1.0. Several other tools have been ported as explained in the appendices.

Our conclusions in brief are as follows:

1. The administration and content management systems in CHEF/ Sakai are more comprehensive than other frameworks such as GridSite as is the suite of tools collaboration available.
2. CHEF is widely used for both Managed Learning (initially at University of Michigan, but taken up in other US institutions) and for Grid projects (e.g. OGCE, NEESGrid, alliance Portal, BioGrid, CMCS, TeraGrid, DOE Fusion, LEAD);
3. CHEF and its tools will evolve into Sakai;
4. Based on evaluation-tree methodology using criteria reported in the concluding section, CHEF/ Sakai scored highest of the open-source portal content management frameworks tested.

It is our opinion that these conclusions make Sakai a clear potential winner for both VRE and VLE operations. A fully objective analysis of our evaluation is provided in Section 6.

Contents

1	Introduction	1
1.1	Background to Portals and VREs	3
1.2	Portlet Standards: JSR-168 and WSRP	6
1.2.1	JSR-168	6
1.2.2	WSRP	6
1.2.3	JSR-168 vs. WSRP	7
1.3	Content Aggregation: WSRP, RSS, P2P, REST etc.	8
2	Portals and Portlets 2003	8
3	Sakai SEPP Developers' Conference	9
4	Role of Portals in a Virtual Research Environment	13
4.1	The Need for a VRE	13
4.2	Capabilities of a VRE	14
4.3	Developing a VRE	15
4.4	A Service Oriented Architecture	16
4.5	Technical Framework	19
5	Sakai Evaluation including CHEF, OGCE and GridSphere	24
5.1	GridSphere	25
5.2	CHEF	27
5.3	OGCE	27
5.4	Sakai	29
5.4.1	uPortal	32
5.4.2	JavaServer Faces GUI Layer	33
5.4.3	JSR-168	34
5.4.4	WSRP	34
5.4.5	OKI Open Service Interface Definitions	34
5.4.6	Spring	37

5.4.7	Database and Hibernate	37
5.5	Available Portlet Tools	38
5.5.1	Collaboration	38
5.5.2	e-Research	38
5.5.3	e-Learning	39
5.5.4	Digital Information	39
5.6	Outcome of Evaluation Work Packages	39
5.6.1	Evaluation Report Part 1: Technology Survey	39
5.6.2	A Review of the Issues for building standards-compliant Portlets	39
5.6.3	An Assessment of the Potential of Sakai as a Platform for Customised Portals	39
5.6.4	Evaluation Report Part 2: Developer and User Feedback	40
5.6.5	Software Template for Sakai Institutional Adapters	40
5.6.6	Software Template for Grid Tool Wrappers for use in Sakai/CHEF	40
5.6.7	Roadmap for a UK Virtual Research Environment	40
5.7	Our Proposed Development Strategy	40
6	Technology Evaluation Report and Conclusions	41
6.1	Introduction	41
6.2	Method	41
6.3	Score Justification	42
6.4	Combining the scores and weights	46
7	Acknowledgements	46
A	WP 1 – Comparing Sakai/ CHEF with Alternative Frameworks for a VRE	52
A.1	Portals and Portlets 2003	52
A.2	ReDRESS Technology Evaluation Report	52
A.3	Evaluate new Work	52
B	WP 2 – Ease of Administration (EoA) of Sakai/ CHEF for a VRE.	53
B.1	Add Users	58
B.2	Customisation	59

B.3	Worksite creation	59
B.4	Security	59
B.5	Performance and scalability	60
C	WP 3 – Making existing VRE (Grid) Components available in Sakai/ CHEF.	60
C.1	InfoPortal	61
C.2	HPCPortal and NGS Portal	61
C.3	CCF Whiteboard	63
C.4	SPP Cross search	63
C.5	CopperCore Integration	64
D	WP 4 – Issues involved in extending the Functionality of Sakai/ CHEF	64
D.1	Web Services and VRE Extensions to Sakai Framework	73
D.2	Implementation	74
D.3	Technical Specification	75
E	Another independent Technical Report	75
F	Integration of UK VRE Tools into Sakai	78
G	Resources	81
H	Service Capability Set	84
I	Generic Portal Engines	86

1 Introduction

Our primary aim is to evaluate the appropriateness of the Sakai/ CHEF frameworks for the development of Virtual Research Environments (VREs) within the UK education, digital information and research communities. This report should help JISC to formulate the long-term requirements of VREs. The relationship between, and capabilities of, the CHEF/ OGCE and Sakai frameworks is explained in the full report. Another JSR-168 compliant framework, GridSphere, is also described for completeness as it has a certain popularity for Grid projects in Europe.

A broad definition of a VRE was provided following some debate by the JISC JCSR VRE Working Group [45]. It encompasses some, but not all functionality of:

e-Research: access to research tools, personal information, project-related issues;

e-Collaboration: contact with and working with researchers in the same or related fields;

Digital Information: access to relevant resources to provide background and supporting evidence for research and training;

e-Learning: components relevant to awareness and training. We assume that more research-related learning will be self-motivated and guided by example rather than managed coursework. Assessment tools are therefore ignored;

e-Management: project and financial management in a research environment;

e-Authoring: authoring of all project and related research oriented material, often requiring input from partners and review by peers and champions;

e-Publishing: publishing in appropriate format of research outcomes. Encompasses the peer-review process;

e-Leisure: BBC news, weather, current affairs, finance, local issues and events.

CHEF, the underlying technology which was the initial focus of this review, is an open source, freely available framework designed to provide useful collaboration tools, such as a chat facility, discussion boards, shared calendars and file sharing in the context of a course or tutorial 'worksite'. It currently provides analogues of the most popular features of commercial groupware offerings such as IBM's Lotus Notes. The Sakai project aims to bring CHEF and uPortal together and then add more VLE facilities. It is in fact designed to encourage open collaboration and the sharing and contribution of new tools by a large and growing community of users, and this appears to be appropriate to the needs of the UK research community. After some discussion it was felt appropriate to include GridSphere in the evaluation of open source tools as it is being used in some UK e-Science projects. Bodington was included for completeness because of its prominence as a UK Virtual Learning Environment.

Jetspeed and uPortal are open source, freely available portal servers that allow portal content to be imported, or linked to, by an administrator and then flexibly configured by users, thus empowering the users in the development of their own collaborative environment. They are hosting containers supporting portlet standards, but do not offer higher-level content management or tools and therefore are not included in the final assessment. Other generic portal engines are listed in an Appendix.

Virtual Research Environments, VREs, by their very nature will continue to evolve. It is important to make sure that our existing resources, services and applications can be made accessible in emerging standards-compliant frameworks and can be supported in the long term. A part of the funding requested for this evaluation was for a UK subscription as early adopters to the Sakai educational Partners Programme, SEPP. There are obviously good financial reasons for moving to an open source platform for VRE delivery and

collaboration, such as Sakai/ CHEF. The software is free and works on various platforms, being 100% Java code on the server side. There is thus no tie in to specific hardware, and the system will scale well financially with no licensing costs for extra server nodes etc. There are good logical and technical reasons also. Sakai will be completely open source and open architecture, so institutions will be able to customise existing, or add new, Java JSR-168 compliant portlet codes to connect to a legacy system running on their campus, e.g. to use existing project services such as databases, evaluations and timetables. Sakai will be configurable to use various different database management products, both commercial and open source, thus avoiding RDBMS lock in. At Daresbury an Oracle 9i RAC meta-data server and SRB are being used, but not all projects use Oracle so we tested PostgreSQL too. By exposing existing VRE services using the standard portlet API through a customisable portal framework, and re-using much existing code and sharing additional collaboration tools, we can achieve a large reduction in software development outlay and encourage closer community integration. Of course, this reduction has to be offset against the potentially increased need for software developers to throw away their bespoke solutions and adapt or interface to legacy systems so that they can co-exist with a VRE framework. If they do so however we can achieve a portable and maintainable solution.

This evaluation has addressed the following areas:

1. Comparing Sakai/ CHEF with Alternative Frameworks for a VRE. WP 1 is reported in Appendices A and B (the latter re-produced in the separate summary document);
2. Assessing the Ease of Administration (EoA) of Sakai/ CHEF for a VRE. WP 2 is reported in Appendix C;
3. Establishing the feasibility of making existing VRE (Grid) components available in Sakai/ CHEF. WP 3 is reported in Appendix D;
4. Establishing the issues involved in extending the functionality of Sakai/ CHEF particularly to use Web services for distributed development and deployment. WP 4 is reported in Appendix E;
5. Developing a *Roadmap for a UK Virtual Research Environment*. This is now in a separate report from the JCSR VRE Working Group [45].

The deliverables of the project summarised in the remainder of the full report report are:

1. Evaluation Report part 1: Technology Survey (WPs 1,2);
2. A Review of the Issues for Building Standards Compliant Portlets (WPs 3,4);
3. An Assessment of the Potential of Sakai/ CHEF as a Platform for Customised Portals, e.g. ReDRess, NCeSS, e-HTPX, e-Minerals and NGS (WPs 3,4);
4. Evaluation Report part 2: Developer and User Feedback (WPs 2-4);
5. Software Template for Sakai/ CHEF Institutional Adapters (WP 2);
6. Software Template for Grid tool wrappers for use in Sakai/ CHEF (WPs 3,4).
7. *Roadmap for a UK Virtual Research Environment* [45]

Our work has been written up in this report and also included in a couple of papers to the 2004 e-Science All Hands workshop, see References [30, 31]. An interim report was submitted to Nicole Harris of JISC on 3/9/04.

Separate summaries, project deliverables and the full report are available from the *Sakai Evaluation* worksite of the ReDRess Portal. It is accessible from <http://redress.lancs.ac.uk:8080/portal> by logging in with (username=guest, passwd=eResearch). The full report is also available from http://www.grid5.ac.uk/Sakai/sakai_doc.pdf.

- *SAKAI EVALUATION EXERCISE* Rob Crouchley, Adrian Fish, Rob Allan and Dharmesh Chohan (this report)
- *SAKAI EVALUATION EXERCISE - Summary* Rob Crouchley, Adrian Fish, Rob Allan and Dharmesh Chohan
- *SEE: Architectural Summary* Rob Crouchley and Rob Allan
- *SEE: Technical Summary* Adrian Fish and Dharmesh Chohan
- *SEE: Interim Report* Rob Crouchley, Adrian Fish, Rob Allan and Dharmesh Chohan
- *Sakai Installation Guide* Adrian Fish
- *Sakai Architecture Evaluation* Adrian Fish

It is clear that, whilst we have attempted to be inclusive in our survey and review, such work is time-bound and never complete. We apologise if we have omitted to reference the full set of appropriate tools and technology and beg their authors to contact us with more information.

1.1 Background to Portals and VREs

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.

At the *Portals and Portlets 2003* workshop [28], 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 [32, 33]. 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.

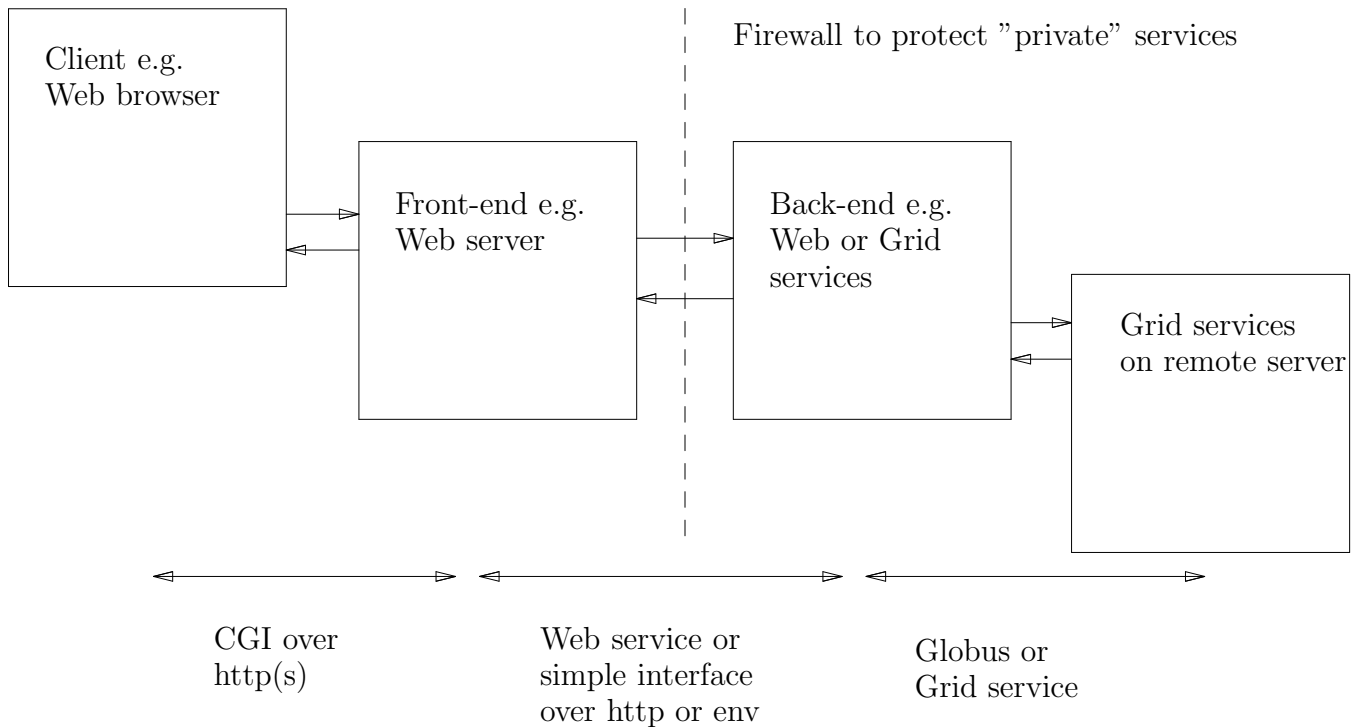


Figure 1: Four-stage Tool Architecture

Figure 1 shows a simplified picture of a fairly typical architecture for Grid-based tools as used in HPCPortal from CCLRC [8]. 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 desirable 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;

Remote resource: accessed from a back-end service by a remote procedure call using Globus or other Grid middleware such as LSF, Condor or PBS 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 services are needed in an Open Services Architecture [44, 55, 71]. 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 underlying 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 extensibility 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 presented 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 environment
- 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.

1.2 Portlet Standards: JSR-168 and WSRP

1.2.1 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.

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 I in Appendix I.

1.2.2 WSRP

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

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 and Sun have taken 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 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>.

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 as a WSRP service either via a portal framework or by a WSRP4J wrapper in a UDDI registry and import it into another portal using a portlet proxy or WSRP4J. See [68] and [69].

We note that WSRP services can also be consumed in different ways, for instance it is possible to use and portal framework or to use Swing [70] to render portlet services in a GUI.

1.2.3 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, uPortal) whilst WSRP specifies the interfaces for accessing portlets across portal frameworks. 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.

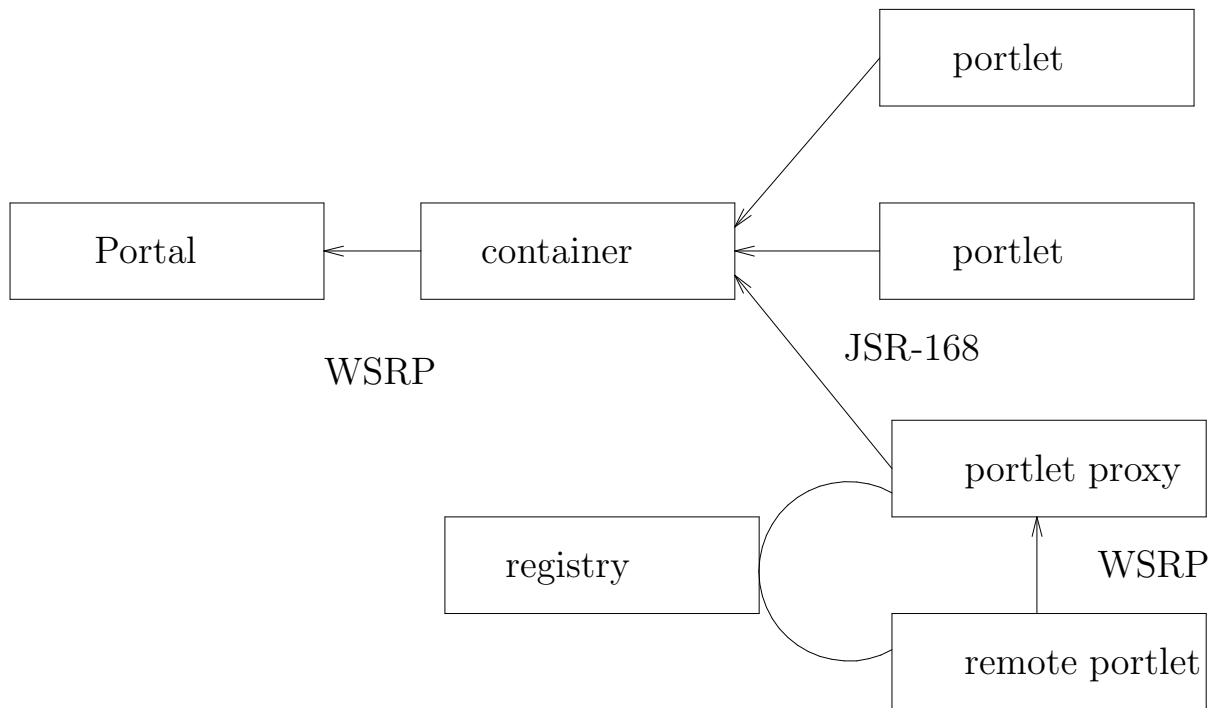


Figure 2: Relationship between WSRP and JSR-168

It is important to note that you don't need a portal framework 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. This is also an important aspect of wrapping more traditional applications for presentation via a portal framework. For Java programmers, WSRP4J [69] can act as a bridge either to publish a JSR-168 compliant portlet as a Web service or to ingest the WSDL referring to a WSRP service and produce the required JSR-168 interface to plug into a portlet container such as uPortal, Jetspeed or LifeRay.

1.3 Content Aggregation: WSRP, RSS, P2P, REST etc.

A number of documents are available describing uses of aggregation protocols applicable to portal environments. These have been used in the evaluation exercise. In particular one describing the use of WSRP in Web service registries such as UDDI is of interest because it indicates how remote portlets can be published and discovered in a similar way to other services. This was discussed in a Grid context at a recent OASIS workshop (Universit of Oxford, 20-21 September 2004). Proposed specifications are due for public review in October, see [68].

2 Portals and Portlets 2003

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. The workshop *Portals and Portlets 2003* was held at the National e-Science Centre, Edinburgh, 14-17th July 2003.

The agenda and presentations from the workshop, including the majority of presentation materials, are available on line at the NeSC Web site <http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=261>. Links to portal projects are included in the full report [28].

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.

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.

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. There have recently 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 workshop conclusions as follows.

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 recommended 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 *UK e-Science Gap Analysis* [34];
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 seemed 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 did appear to be a convergence in technology and benefits from sharing some of the tools being developed. To this end we looked 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. Conclusions are noted in this report of the Sakai Evaluation Exercise.

3 Sakai SEPP Developers' Conference

The authors of this report attended the first Sakai Conference in Denver, Colorado 23-27th June, 2004. Other UK participants included Sarah Porter (JISC), John Norman (University of Cambridge) and Ian Dolphin (University of Hull). This meeting was also the inauguration of the Sakai Educational Partners' Programme, SEPP.

The SEPP includes resources for community development, training, shared best practice, and also early access to Sakai releases. The SEPP goals are:

- To involve a wider community in the Sakai Project's open source vision, economics, and innovation;
- To mobilize distributed resources for development and support of Sakai tools;
- To initiate a "market of tools/ components" that will run in the Sakai framework;
- To manage interaction with the Sakai Core development team;
- To coordinate activities with other organizations, such as IMS or country-level agencies;
- To build on the experiences of the JA-SIG, CHEF and OKI training and conferences;

- To generally engage in Sakai community development of shared best practice.

The majority of the participants were already SEPP members. Over 175 people from 45 institutions attended tutorials, presentations, meetings, and BOF sessions. The enthusiasm for the Sakai project was leavened with some hard hitting technical questions and concerns for the future. The Sakai Board of Directors, led by Joseph Hardin, emphasised that SEPP has a vital role to play as we move into Sakai 2.0 development and beyond.

The current list of SEPP members includes:

Arizona State University
Boston University, School of Management
Brown University
Carlton College
Carnegie Foundation for the Advancement of Teaching
Carnegie Mellon University
Coast Community College District (Coastline Community College)
Columbia University
Community College of Southern Nevada
Cornell University
Dartmouth College
Foothill-De Anza Community College District
Georgetown University
Harvard University
JISC (UK)
Johns Hopkins University
Maricopa County Community College District
Nagoya University
New York University
Northeastern University
Northwestern University
Ohio State University
Princeton University
Simon Fraser University
State University of New York
Tufts University
Universitat de Lleida
University of Arizona
University of California, Berkeley
University of California, Davis
University of California, Los Angeles
University of California, Merced
University of Cambridge (UK), CARET
University of Cape Town, SA
University of Colorado at Boulder
University of Delaware
University of Hawaii
University of Hull (UK)
University of Oklahoma
University of Virginia
University of Washington
University of Wisconsin, Madison
Virginia Polytechnic Institute and State University
Yale University

The Denver conference included a number of important presentations from the Sakai Core Team as well as from SEPP members who had already begun to develop and port educational tools into the Sakai framework. We here focus on points raised in the core presentations. Most of the presentations shown are available on the SEPP Community portal site. A Syllabus article covering the conference is also available at: <http://www.syllabus.com/article.asp?id=9635>. A Chronicle of Higher Education article covering the conference is available too: <http://www.newswire.com/articles/view/505760/> written by J.R. Young. A longer list of relevant articles, particularly concerning its use in education, is now available on the Sakai Web site.

The Sakai Tools Team had released three documents related to Sakai functionality. The first document covers Samigo (based on tools SAAM and Navigo) and its feature set:

http://chefproject.org/access/content/group/1075771392979-922/SEPP_Conf_June_2004/Samigo\Marketing.doc

A second document covers general Sakai functionality:

http://chefproject.org/access/content/group/1075771392979-922/SEPP_Conf_June_2004/Sakai\Fall104\mktg.doc

In addition, here is a link to the 15MB PowerPoint presentation that explains each gap in great detail as was promised at the SEPP Conference:

http://chefproject.org/access/content/group/1075771392979-922/SEPP_Conf_June_2004/Gap_Prioritization_Revised2.ppt

SEPP Requirements Group met at the SEPP conference with the BOF on content and authoring and later on its own. Some of the key areas that it identified include:

- Get more institutional input into the high-level build list (match institutions to tools) to contribute to the 3.0 product;
- Matchmaking to encourage *ad hoc* alliances to form around tool development and common areas of interest;
- Promote a common specification and development process;
- Provide links to the core teams and ease communication efforts;
- Provide a common repository for discussion group requirements and specification documents;
- Provide a gap analysis between what is currently in 1.0 and 2.0 and the stated partner requirements;
- Encourage partners to become more forthcoming with what they are doing via a suggestion area;
- Provide a process for requirements feedback for projects to encourage a broader understanding of common needs across institutions;
- Keep a running list of current projects, participants, and project information.

Some thoughts were shared during the conference regarding project management within *ad hoc* alliances:

- Mutual interest and timing (alignment) is critical;
- Create mini-projects using the SEPP process template;
- A project leader should be assigned from one of the participating departments who will be responsible for staff resources across institutions and management of project;
- Projects need to have visibility via Sakai forums.

We in the UK are recommending the formation of a Sakai Alliance which would work in this way and follow the SEPP guidelines.

Challenges identified:

- Maintaining openness and communication while keeping tasks on track;
- Ensuring visibility of projects and crediting their authors;
- Maintaining a process where people understand where and how they can participate (especially if projects

are already underway).

The requirements group needed volunteers willing to put in some time and effort to creating the necessary matchmaking tools, compiling the build list, and performing a gap analysis.

SEPP Development Group. On Wednesday, Mark Norton gave an overview of the Sakai architecture and how to write tools using the Sakai framework and Tools Portability Profile (TPP).

The primary contributions to the Sakai tools project include the following.

- The first contribution will be the Technology Portability Profile that will describe the integration of OKI's OSIDs, a user interface abstraction for localization and the new JSR-168 portlet specification. This Profile integrates the successes and lessons of Michigan's CHEF interoperability framework, and JA-SIG's uPortal, and describes a common path forward for their respective developer/ user communities;
- The second contribution is that Michigan, Indiana, MIT, Stanford, and uPortal will all license their intellectual property and/ or experiences with large scale application software (e.g., Course Tools, Work Tools, Navigo Assessment, Oncourse, Stellar, uPortal, OneStart, Eden Workflow, CourseWorks, etc.) into a re-factoring of best features. This will include an enterprise-scale course management system, distributed research collaboration tools, and an enterprise services portal (described in more detail below), and others that have been conformed to the Technology Portability Profile. All Sakai tools will be both modular and also pre-integrated to work with each other. The software will be made available to the world at the same time via an open source license;
- The third contribution overcomes the barrier of institutional timing by synchronizing the development and implementations at four large institutions: Michigan, Indiana, MIT, and Stanford. All institutions are committing to an initial implementation of the Sakai tools, as a campus-wide course management system and/ or campus-wide enterprise Portal by 3Q05 when the tools are fully released. Synchronization will greatly facilitate further shared developments in the following years.

Secondary contributions were deemed to be no less valuable and in the longer run may be more valuable to the Sakai user communities.

The second evening of the conference comprised of an "All Hands" style demonstration session exhibiting the range of tools which are now being ported to Sakai by the SEPP members. This list was impressive:

Tufts	VUE
Virginia	Fedora
Rsmart	OSPI
Wisconsin	Virtual Dig
Wisconsin	Lesson Builder
Northwestern	Annotation tool
Indiana	Sakai Assessment Manager
University of Toronto	TILE and Atutor
Northwestern	Project Pad
University of Michigan	Sakai
Indiana	weBWorK
Knowledge Media Laboratory	KEEP
Tufts	TUSK
Harvard	iSites and Videotool
UC Davis/ Princeton	Image Management System
SUNY	Learning Environments
OKI OSIDs	Digital Repository

Our questions to the developers and SEPP members included:

- OKI – what is it all about and how is it being used in Sakai?
- Extending the API – are our suggestions worth pursuing?
- Contributing – how can the UK developers best do this and be taken seriously?
- uPortal involvement – how extensive is this and what are the plans and timelines, e.g. for JSR-168 and WSRP?
- the SEPP partners – who are they and what are they doing? Can the UK partners identify niche roles?
- e.g. UK role: WSRP and tools – we discussed this with Chuck Severance, Mark Norton and others and identified some early desirables such as a Wiki tool and WSRP demonstration.

4 Role of Portals in a Virtual Research Environment

There were a number of fundamental research questions which must be answered before the large-scale deployment of a Virtual Research Environment in the UK. Our Sakai evaluation and additional work with the JISC JCSR VRE Working Group have lead us to identify and suggest responses to many of these issues. Portals are just one, but an important, component of a VRE.

4.1 The Need for a VRE

e-Science is a new paradigm of research, often characterised by a “deluge” of data analysed by massive distributed computing power. e-Science research collaborations are frequently large, distributed and multidisciplinary involving hundreds of institutions across the globe. Grid technology, emerging in response to these challenges, is enabling exciting possibilities for better research, even creating new disciplines like astro-informatics. In this context, a wide range of national and international initiatives are under way.

The concept of e-science is now broadening and evolving into e-research generally, to encompass the social sciences and the arts and humanities. At the same time it has to be recognised that different communities are at very different stages in their awareness of the new technologies: thus the current needs of a large international scientific collaboration are likely to be much more complex than those of the lone humanities researcher, wishing to collaborate more effectively with a handful of colleagues world-wide in the same field of interest. In our thinking we have tried to keep the whole range of requirements in view.

At the high end, the new developments are making the process of carrying out research more complex and demanding. The aim of a Virtual Research Environment (VRE) is to help researchers manage this complexity by providing an infrastructure specifically designed to support the activities carried out within research teams, on both small and large scales. JISC has recently been allocated £3.2 million as part of the Comprehensive Spending Review to develop a VRE.

The challenge is to create and sustain an infrastructure, ideally usable on a routine basis by researchers from all disciplines to enhance their productivity and effectiveness. Meeting this challenge is a task for those building the infrastructure, its potential user communities, the institutions to which users belong, the organisations which fund research and other stakeholders in the research process. These developments should not happen in isolation but will need to interwork with other components of the infrastructure being provided by JISC, Research Councils and HEIs themselves such as learning environments, digital libraries and national research facilities.

It is unclear, a priori, what type of framework a VRE should adopt, on which technologies it could be based, how it can be developed sustainably and how usability and take-up can be ensured. This document outlines a roadmap for developing a VRE. The target user community is all those engaged in research.

Disciplines, and communities within disciplines (especially in non-scientific subject areas), will have to identify the possibilities for them in the technology; may have to overcome cultural obstacles to collaboration; and may need training in relevant skills. Associated legal issues will need to be understood and clarified, and formal and informal codes of practice updated, to reflect understanding of novel forms of collaboration.

Locally, institutions will need to understand the business case for supporting research collaborations and how they can be reconciled with continuing institutional competition. Wider impacts will be felt through changes in scholarly communication and in the complexities of managing and sustaining long-term open access to data for reuse.

Additional background material and rationale for creating a VRE is contained in a report entitled *Building Collaborative eResearch Environments*, compiled for JISC by Andrew Cox, Department of Information Science, Loughborough University [41]. This summarises the proceedings and breakout group discussions from two workshops held in March/ April 2004 at Edinburgh and Warwick universities. The report also contains a SWOT analysis. A brief summary of some of the main recommendations from these workshops is included as Appendix A of the report.

Background material from the UK e-science Grid, compiled by the Architecture Task Force, is contained in the report [42] *UK Role in Open Grid Services Architecture* by Malcolm Atkinson et al. This vision is being realised through the work of the Engineering Task Force and is being introduced onto the production National Grid Service (including the JCSR-funded computing and data clusters) during 2005.

Related JISC work on e-learning frameworks and tools is described at http://www.jisc.ac.uk/index.cfm?name=elearning_framework [43]. Work on developing a distributed architecture as part of the JISC Information Environment is described at <http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/> and in reference [44].

4.2 Capabilities of a VRE

The nature of a VRE means that it is more realistic to describe it in terms of its intended capabilities rather than its component parts as the latter are likely to evolve over time, depending on contemporary standards.

In the views of the JCSR Working Group, a VRE should:

1. Support the processes of conducting research, including marshalling of resources, scholarly discourse and publication, and the creation and maintenance of collaborations, across disciplines, institutions and countries, including support for meetings and organisational processes.
2. Be based, as far as possible, on loosely-coupled, distributed, interoperable tools, rather than a monolithic piece of software.
3. Be designed to meet user requirements and address usability and accessibility, with appropriate evaluation mechanisms and benchmarks for new tool development.
4. Include modes of access which (almost) any user can download and install on their laptop/ desktop/ PDA/ home computer, with “servers” that can easily be installed by system administrators without specialist knowledge and national JISC-provided servers as appropriate, so that tools work “straight out of the box”. Some tools will be integrated with domain-specific facilities (and vice versa).
5. Adopt and use appropriate open standards wherever possible.

6. Be secure and trustworthy. Hence the VRE components should interoperate with federated cross-institutional authentication and authorisation mechanisms.
7. Be accountable, by providing adequate logging and probity including supporting queries about provenance.
8. Be compatible with other widely used and deployed systems, including at least: web, email, instant messaging, SMS, Wikis and videoconferencing tools from lightweight desktop applications through to high-end videoconferencing via Access Grid. This means that the VRE should be accessible via web browsers and 3G mobile phones among other modes of access.
9. Support creation, sharing and curation of resources, through ease of authoring, publishing, discovery and access. This implies adoption of appropriate metadata schema and support for automatic generation of metadata. Resources to be described will include data, computation and potentially humans.
10. Be extensible with enhanced or new tools by any developer, through use of published standards and provided software development kits, software libraries etc. It should be as easy as possible to make existing software and services (e.g. e-print repositories, portals), including proprietary software, compatible with the VRE.
11. Be open source and standards-compliant wherever possible. The licensing of the tools should encourage and support improvements to the tools and development of new tools through open source development by the community.
12. Support tailoring of the environment by individuals or groups to reflect their interests and preferences.
13. Support the delegation of routine tasks to intelligent personal agents where the means to realise these exists, e.g. by incorporation into workflow processes.

4.3 Developing a VRE

Based on the background and examples summarised above, a VRE can be considered as a set of applications, services and resources integrated by a standards-based, service-oriented framework which will be populated by the research and IT communities working in partnership. The scope of the components needed to build this framework is further discussed with examples in Section 5.

Multiple domain-specific or community-specific gateways to the VRE will exist, in parallel serving the needs of different communities but achieving maximum synergy and cost-effectiveness by being based on a common framework which enables reuse of generic open-source components, referred to as services. The example of portals given above is just one example; lightweight programming libraries is another, permitting integration into “heritage” codes.

It is not the intention of JISC’s VRE programme to produce a complete VRE, but rather to define and help to develop the common framework and its associated standards and to encourage others to work within this framework to develop and populate VREs with applications, services and resources appropriate to their needs. The intention is to maximise the value and benefit of future investment in this area by both JISC and the Research Councils, to secure community contributions and to promote sustainability.

VREs must cater for a wide range of scale and complexity of research activities, from small research collaborations with a few partners to large teams with many partners in many institutions. Examples of the latter include projects within the current e-science programme. A balance must be achieved between meeting the needs of specific disciplines and developing capabilities of widespread utility. The wide variety of research activities means that a judgement must be made in each case on the appropriateness of including specific capabilities for the application in question.

In the long run, VREs will have to become self-sustaining within their user communities and service providers. They will therefore have to be seen to provide sufficient additional benefit to motivate this effort. Whilst VREs, gateways and resources will "belong" to their user communities and will respond to and track these communities' evolving requirements, it is expected that there will be mutual benefit in coordinating these separate VRE activities through a common framework with re-useable services and associated standards. JISC, in its closer relationship with the Research Councils, may continue to provide a suitable common reference point for this coordination. There is potentially a role for the Open Middleware Infrastructure Institute (OMII) in providing ongoing support for VRE middleware.

It is important that the VRE development activity has strong links with other related activities such as JISC's VLE and MLE programmes, the research programmes and communities supported by the individual Research Councils, and international developments such as Sakai in the US. It will have many components and services in common with these.

4.4 A Service Oriented Architecture

A Service Oriented Architecture (SOA) is an approach to joining up services to provide integrated capabilities. It is a relatively new approach, but is rapidly gaining popularity because of the lower costs of integration coupled with flexibility and simplified configuration. This is becoming best practice for commercial distributed software development, see recent reviews e.g. [47, 48, 49, 50, 51]. An SOA builds upon the use of web services, the emerging industry standard for building and integrating distributed systems. The rationale for using an SOA in the JISC context for MLEs/VLEs is given in [43]. Other relevant projects worldwide are considering and indeed beginning to deploy similar approaches and architectures. One worth noting is Arda, the next generation framework for distributed analysis of Large Hadron Collider data [52].

Portal deployments have, until now, typically been monolithic with a rich set of tools, customisation possibilities and a database for content management. CHEF, OCE and Sakai fall into this category and are deployed as large Java jar files. At CCLRC the Integrated e-Science environment, IeSE, comprising services from HPCPortal, InfoPortal and DataPortal has already tried to break this mould [16]. Another more recent activity at University of Indiana is taking this a stage further and now using CHEF [54]. In Section 6 we propose an architecture for presentation of services through portlets in an SOA which extends these ideas.

The following figures highlight some basic aspects of an SOA relevant to deploying a VRE with appropriate user interfaces such as portals, online commands, drag and drop desktops and programming libraries. A key aspect of the architecture is to maximise the re-use of common services and middleware including portlets.

Figure 3 shows how an SOA approach would be of benefit in exposing a common set of services and middleware through a variety of user interfaces including Web portals employing the WSRP standard. It indicates how this architecture can be used to facilitate the horizontal aggregation that can occur for specific groups, e.g. the National Centre for e-Social Science (NCeSS) which is working alongside the Lancaster node for Quantitative e-Social Science (CQeSS) and the JISC/ ESRC training and awareness programme ReDRESS, see <http://redress.lancs.ac.uk>.

An SOA clearly does not preclude also using portals or data warehouses, and is in fact agnostic about how the rest of the enterprise is configured, which is why it makes a good approach for a framework. In addition, because integration occurs in this fashion, it becomes a simpler task to replace the systems that provide services within the architecture or to look up new ones via a registry such as UDDI [56]. Because service consumers are configured to access a service without any knowledge of the system that provides the service, we can replace the underlying system without affecting systems dependent on its capabilities.

Figure 4 shows how services are used in a typical 4-layer architecture for portals and other client tools.

Notes:

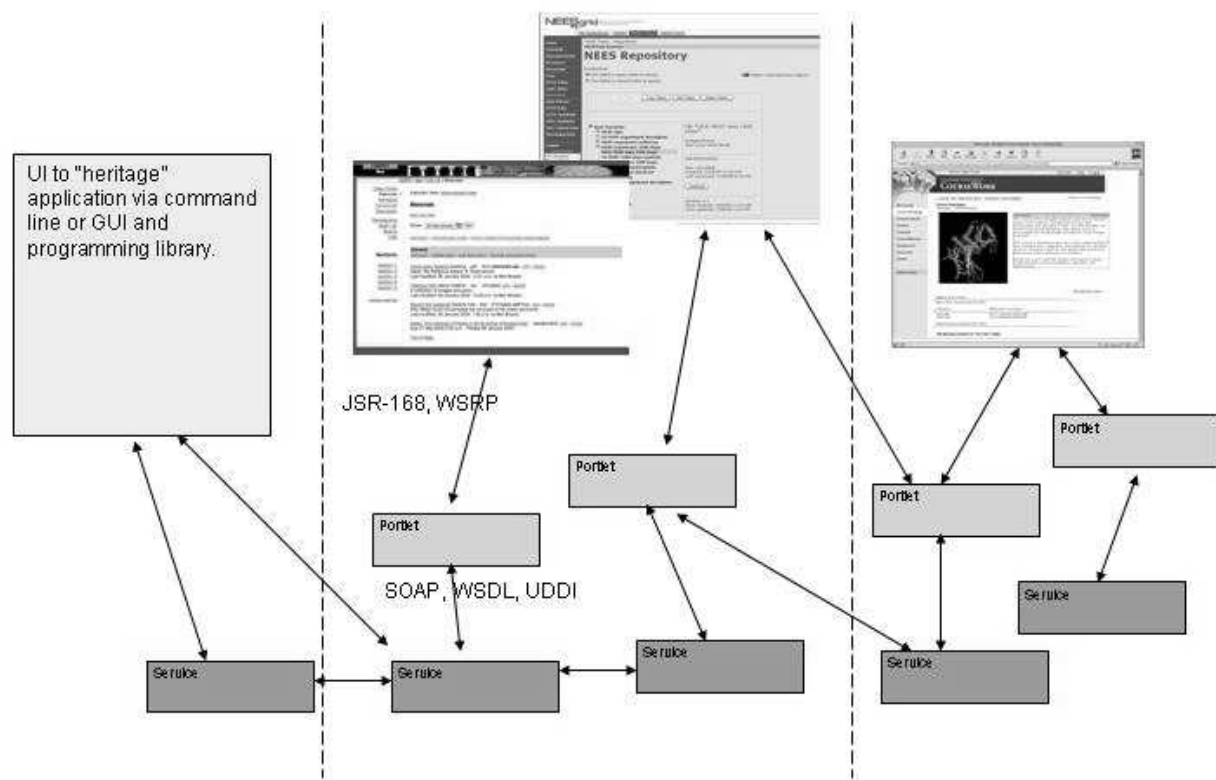


Figure 3: The Grid of Services

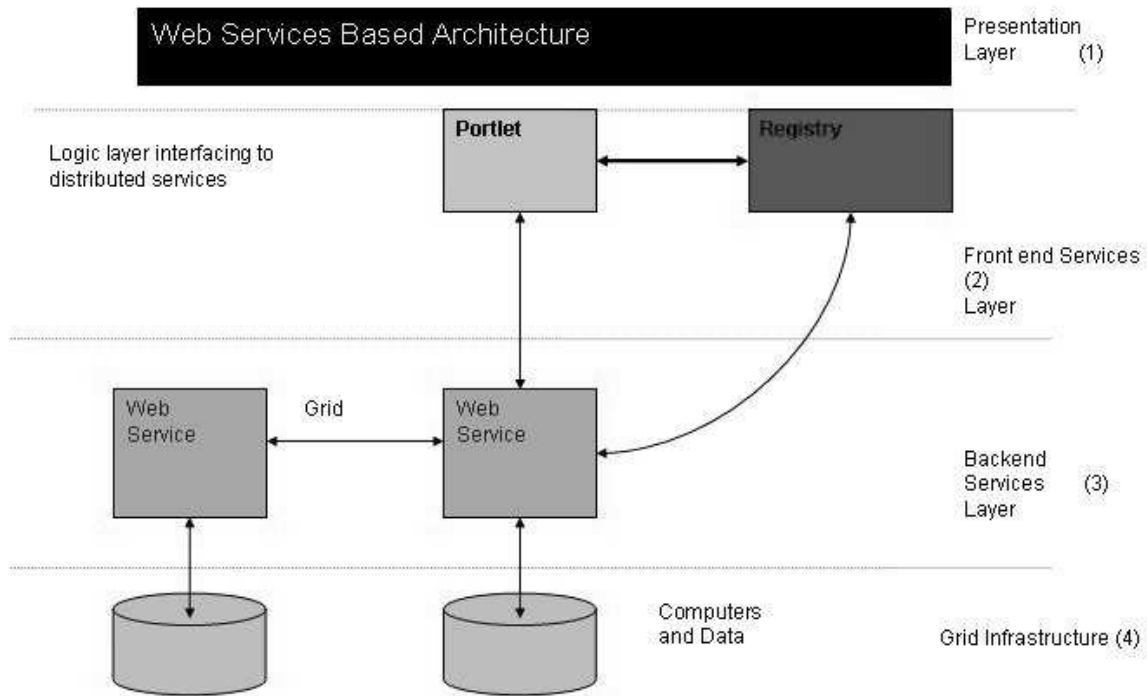


Figure 4: 4-layer Portal Architecture employing Web Services

1. Presentation layer, e.g. uPortal, Jetspeed or similar integration and rendering framework. Tools such as Sakai/CHEF add value to this by providing a content management system and additional service interfaces.
2. Front end services, logic layer interfacing with (1) through the JSR-168 portlet API.
3. Back end services, accessed via web service or other well-known protocols: could be distributed, and/or could interact with other web services forming a Service Grid.
4. Remote resources on a Grid infrastructure, e.g. computers and data bases, such as the JCSR clusters.

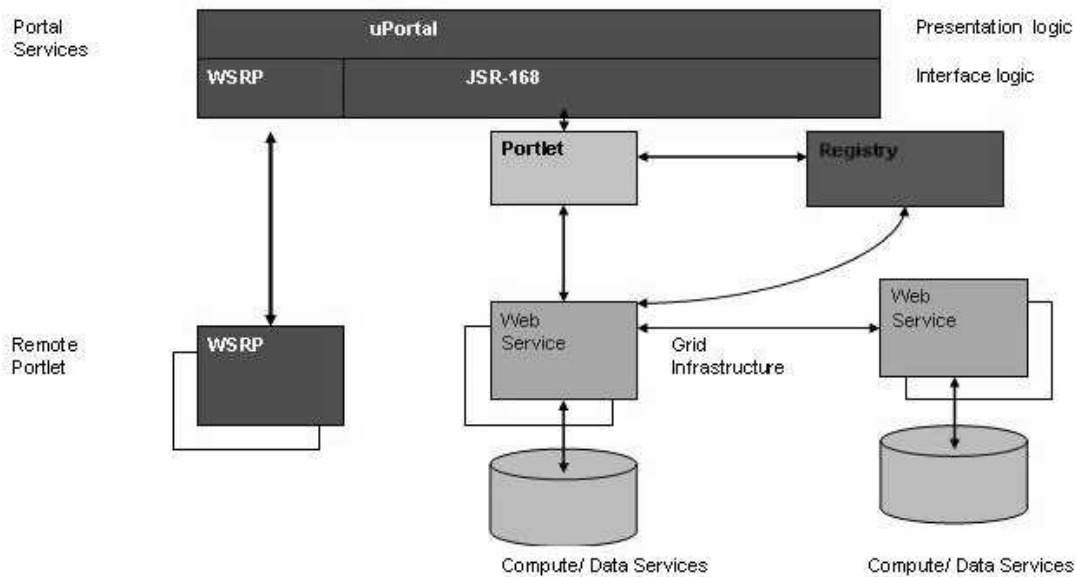


Figure 5: Portlets using Web Services

This is further discussed in Appendix D.

4.5 Technical Framework

This section briefly outlines technical aspects of a VRE which may indicate the scope of work required. This discussion arises directly from the identification of an SOA as the preferred architecture for the VRE and

the community input which has led to the workshop recommendations. It is easy to be all-inclusive in a theoretical approach, but a more carefully chosen set must be suggested for pragmatic reasons. The following sections therefore attempt to suggest an initial preferred set.

Frameworks/ Interfaces

Following discussions among interface providers, such as those writing 2nd Generation Portals and web services based middleware, we suggest adopting the following standards:

- JSR-168 (Java standard for portlet interface);
- Web Services (WSRP, WSDL, WS-I) which may extend to such things as WSRF in future (SOAP over HTTP is mainly implied as a request/response protocol but other delivery protocols may be needed, e.g. for large datasets, see below);

Frameworks should therefore be able to support these interfaces, e.g. in the case of portals Websphere, uPortal, Jetspeed, GridSphere etc. are possible candidates. For further discussion on JSR-168 and WSRP see [28]. There will be multiple solutions for access to the VRE and standards and inter-operability are key issues. In addition to frameworks such as portals, services must be able to be incorporated into heritage applications and GUI interfaces such as those developed and maintained by the long-standing Collaborative Computational Projects (CCPs), which are a unique contributor to UK research output.

Generic Services

These generic services are an abstraction of actual services, some of which might be made available via the VRE. Appendix H provides a fine-grained, but still evolving, list of services from areas such as:

- AAA, Grid Services, Semantic Services, Resource Discovery, etc.
- E-Collaboration;
- Support and Management Services.

This is an attempt at an initial classification and description of these services based on work of other groups such as JCLT, ETF and GGF. Further work is required to identify how services can be broken into methods (functionality) and if appropriate where existing middleware provides such functionality. See <http://www.grid.ac.uk/ETF/public/WebServices/classes.html>.

Real Resources

Services must clearly be mapped onto real resources to be of value. In Appendix G we have listed some resources and services already available to the UK academic research community which could be accessible. This list is also not exhaustive, but we should identify the main resources and facilities which are likely to be available via the VRE. It may be the case that some of these resources need extra effort to provide interfaces or full availability. Gaps may also be identified as we analyse these resources and services.

Requirements and Use Cases

Development of requirements is an ongoing process and the awareness days and demonstrations which have taken place so far have informed this document. Following further demonstrations of VRE prototypes additional requirements will emerge as people change their ways of working, just as the invention of the elevator by Otis made the upper floors of skyscrapers into desirable real estate.

As an example of how requirements could be used to inform the list of services and identify workflows we firstly give a very simple use case:

```
User specifies required topic;
Uses search to discover location;
Uses authentication and authorisation to access source;
```

Downloads material using appropriate protocol;
 Interacts with it using appropriate tool(s);
 Does research;
 Writes paper;
 Publishes content;
 Attends international conference.

Increasingly more complex and sophisticated use cases can be devised, and this should be done before large-scale deployment. Firstly we show a single-user case combining access to an HPC application on a remote resource with transfer of a large dataset from a data centre.

User logs in to VRE server

- Login is based on e-mail and password
- Login includes using X.509 certificates for Grid resources
- Use Shibboleth or whatever emerges from JISC security middleware projects (future work)

User session established/ managed/ logged/ preferences selected

User selects a "Research Service Package" on the tools menu

- Could involve a number of underlying tools/ services mediated by workflow to guide user through a complex procedure

User clicks "help" button or is provided with guidance in another form

Need to invest in "accessibility" configurations

User performs a query to list available services/ information sources with particular search criteria and semantic support

Service query is sent to Universal Description, Discovery & Integration (UDDI) registry

- Enables dynamic invocation of Grid service
- Service providers can edit, delete or modify published grid services through UDDI interface
- UDDI returns a list of requested services and a URL describing on how to invoke the grid service

Information/ data query sent to cross-search broker and translated to use appropriate metadata via an information service

- URL returned pointing to appropriate sources
- Translation of format may be required through another service
- Authorisation etc. with remote database la OGSA-DAI
- Protocol for download chosen, e.g. GridFTP

Clicking on the URL will generate a Web Services Description Language (WSDL) file or initiate download by pre-defined method

WSDL converted "on the fly" to a web form and presented to the user

User enters the appropriate additional details and submits the job

- Data transmitted to host system
- Job submission done using Web services and job logged
- Monitoring of job done for future inspection
- Notification of completion or other "event" by e-mail or message board
- Collection of output, visualisation etc.
- Repeat job
- Send output to a different task in a pipeline

Finally an example of using a VRE for collaborative working between two or more users.

These steps could be part of a more complex workflow which could be modified with different data/ components and/ or repeated.

User has option of saving, deleting, printing or visualising the output and making decisions during the workflow session.

Registry of services and data needed and tools to query, find and consume them.

Security (authentication), confidentiality, privacy, accessibility and authorisation mechanisms essential.

Need appropriate presentation to user in "familiar" environment, e.g. Web portal.

User logs on as above
 - Private session managed as above
 Determines if "peers" are present or if there is a hierarchical activity group
 Asks to join group and is admitted
 - Peer session established and managed when more than one participant
 Discourse with or share material with group
 Use collaboration tools:
 - Whiteboards, visualisation, etc.
 - Chat, messaging, resource box
 - Controlled vocabulary
 - Decision support
 - Support for inter-disciplinary research
 In research this is an important element of planning and project management

Protocols/ Standards

To make use of resources within the VRE additional protocols and standards are required over and above the access protocols identified above (e.g. Web services). Obvious requirements include the transfer of large quantities of data between data stores and computers, which requires optimal utilisation of bandwidth via a tuned FTP service. Search and content feeds also amplify the requirements. The following list may need to be extended, but should be contained to a manageable set implemented within the VRE and able to be consumed by delivery mechanisms such as portlets:

- Protocol examples: FTP, VIC, RAT, Z39.50, SRW/ SRU, OAI-PMH, RSS, OpenURL;
- Standards examples: IMS, Dublin Core, LOM, OAI, OKI, etc. including those as appropriate from W3C, IETF, OASIS, GGF.

Criteria for Success

In accordance with the discussion set out in the text, we suggest that a successful VRE would:

- Be applicable to all disciplines;
- Integrate e-research, e-learning and management of digital information to add value to all application areas;
- Be built to current specifications, standards and technology, in order to minimize development costs and time;
- Make UK services and resources available in familiar environments e.g. typically via a web browser;
- Have fast-track links into existing tools, services and resources? some of which have been costly to produce and thus should be re-used;

- Enable an open community process for producing and consuming services and tools;
- Offer choice in presentation, delivery, and service and resource provision;
- Provide maximum ease of use, especially to new users, to optimise take-up;
- Demonstrate added value with respect to existing workbenches, toolsets, portals etc.

Call for Proposals

The discussion summarised above led to a Call for Proposals by the JISC JCSR, circular number JCSR(04)/05. This aimed to spend £3.2M of money from HEFCE on demonstrators, tools and components for a UK VRE. The call deadline was 13th August and the proposal evaluation was completed on 10th September and endorsed on 20th September 2004. The successful proposals using Sakai and related portal technology were:

Project	Framework	Community
Sakai Demonstrator	Sakai	To address the requirement for a single point of access to a comprehensive set of Grid and collaboration services in a VRE.
Integrative Biology	OGCE	To develop a VRE demonstrator to investigate the utility of existing collaborative frameworks to support the entire research process of a large-scale, international research consortium.
Silchester Roman Town	Sakai	To develop a system to facilitate rapidly developing and iterative archaeological research by synchronising the three processes of gathering information, co-ordinating expertise, and managing the resulting body of data.
ELVI	SunGard SCT	To produce and demonstrate a practical framework for the effective deployment of a generic VRE.
EVIE	Bodington	To test the integration and deployment of key existing software components within a portal framework.
ISME	GridSphere and Access Grid	To develop and refine the experimental steering process, shared workspace and distributed visualisation into a VRE making them deployable by dispersed teams of instrument scientists, material scientists and engineers in a transparent and robust manner.
IUGO		To develop a proof of concept system to enable the integration of Web-based content (and references to non Web-based content), related to individual conferences and individual sessions within conferences, thus providing a means to provide far greater benefit to the wider research community than is currently available from conference attendance.
Social Science	WSRP, Sakai, DSpace	To provide and support the Sakai platform for large, distributed social sciences research projects. The focus is on activities to investigate the needs of the TLRP researchers and to evaluate the extent to which Sakai meets those needs.
Politics	Sakai, Access Grid	To develop a virtual research and research skills development environment, capable of expansion and of facilitating multiple participation in the rapidly evolving field of the history of political discourse.
GROWL	C Web services library	To build upon the existing prototype GROWL library to produce a truly lightweight extensible toolkit which complements other solutions.
Humanities	Sakai	To build a VRE for the Humanities by investigating how Humanities research can benefit from ICT and by constructing demonstrators in specific fields.
Access VRE	Access Grid	Extend the Access Grid's VRE infrastructure with new collaboration functionalities from the CoAKTinG project.

Cheshire VRE	Sakai and Cheshire III	To identify and integrate a number of tools and technologies for the data grid and digital library communities which will support collaborative e-Research across institutions and domains using Sakai.
CSAGE	Access Grid	To construct and use semi-immersive stereoscopic facilities to create an increased level of “presence” within the Access Grid environment and the recording of the performance, in the same format, within a framework playable by a larger community base as and when required.
CORE	Chandler and Web services	To provide integrated computer support across the research and educational cycles, because these activities are intrinsically coupled as a part of the requirements of the surgeon’s continuing professional development.

The developers in all 15 funded JISC VRE projects, whether or not they are using Sakai, have been invited to inform each other on their development and use of technology and tools, especially if they can be shared to increase the overall value of the VRE programme. To help this process we have set up collaboration tools as follows:

Sakai Portal Worksite: <http://collab.sakaiproject.org> under “eResearch”. You need to register with the portal and then inform Rob Allan r.j.allan@dl.ac.uk who will enable access.

e-Research Wiki: <http://www.grid.ac.uk/eResearch>. Self register. The above information is taken from this Wiki.

e-Research Mail list: e-Research@jiscmail.ac.uk. Self register.

The Wiki and mail list are linked into the Portal worksite.

5 Sakai Evaluation including CHEF, OGCE and GridSphere

In presenting the results of our evaluation it is important to distinguish and explain the relationship between the several separate but converging open source portal products which have recently attracted a lot of attention and a rapidly growing user base.

GridSphere: JSR-168 compliant portal development framework from the EU GridLab project;

CHEF: CompreHensive collaborativE Framework from University of Michigan, principally a collaboration portal hosting work tools and course tools. Uses JetSpeed portal framework;

OGCE: Open Grid Computing Environment portal, now adopted as the portal being supported by the NSF Middleware Initiative (US equivalent of the UK Open Middleware Initiative Institute, OMII). Built on CHEF content management system with Grid computing tools from NIEES Grid, Alliance Grid and NPACI GridPort. Uses JetSpeed portal framework;

Sakai: VLE portal built on CHEF and using OKI OSIDs with re-factored tools from Michigan, MIT, etc. Uses uPortal framework.

Charles Severance, chief architect of the CHEF and Sakai projects, was invited by JISC to participate as a plenary speaker at the two Awareness Raising Workshops on e-Collaborative Environments [41]. We were thus able to have early discussions with him about our proposed work. Previous discussions with others involved in portal developments, in particular Dennis Gannon and Geoffrey Fox of Indiana University, had taken place at the NeSC workshop in July 2003 [28]. Although Charles Severance was unable to attend this

workshop, the developers of GridSphere, Jason Novotny, Mike Russel and Oliver Wehrens, did attend and gave a comprehensive tutorial of their software on the final day.

Sakai was not available to us at the beginning of our evaluation and required the UK to participate in the SEPP (Sakai Educational Partners Programme). We were however already using CHEF in the JISC/ ESRC funded ReDReSS project <http://e-science.lancs.ac.uk/redress> and Sakai is based on a re-factoring of this system. We were able to gain access to the Sakai developers' CVS to obtain the source code in mid June 2004. SEPP membership also permitted us to represent JISC at the Sakai Developers' Workshop, Denver, USA, 23-27th June.

Evaluation of these related portal frameworks has involved deployment as follows:

Site	Framework	URL
Daresbury	CHEF	http://thames.dl.ac.uk:8080:/chef/portal
	OGCE	http://thames.dl.ac.uk:1081/dnmi
	Sakai v1.0	http://trent.dl.ac.uk:8080/portal
	CHEF	http://portal.ngs.ac.uk
Lancaster	GridSphere	Adrian's laptop
	Sakai 1.5 RC9	https://e-science.lancs.ac.uk:8080/portal

We have included comments on the Sakai installation procedure in Appendix B. These implementations are available for ongoing development, evaluation and demonstrations to interested parties. The following descriptions were taken mostly from the relevant Web sites during the installation work.

5.1 GridSphere

The GridSphere project is building on experience in the Java-based Astrophysics Simulation Collaboratory and GPDK portal toolkits. It now has a reasonable uptake in Europe and to some extent Canada.

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) paradigm 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, American, 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 associated class files and JSP pages. Authorised users can deploy portlets dynamically providing Tomcat v>4.1.18 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/ logout Service – Allows a User to be retrieved from a username and password;
- Locale – banners and support for 7 languages;
- User Manager Service – Add/ Remove User Accounts; Edit User Profiles;
- Access Control Service – Add/ Remove User Groups; Add/ Remove User Roles;
- Layout configuration – customise layout by creating new tabs which portlets can be added to;
- 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;
- Text messenger – communicate to IM users;
- Photo album – upload and display photos;
- Poll – create poll and display results;
- Chart – chart service using JFreeChart to display plots, timeseries graphs, etc.;
- Commander – secure portal file system to upload, download and transfer files.

A number of core portlets are provided by which these services can be accessed. Future plans included the provision of a complete general JSR-168 compliant framework (achieved in February 2004), integration of the GridLab GAT toolkit and OGSA Grid services, an IDE, inclusion of Flash presentations and some forms of collaboration tools.

The following table shows some existing projects using GridSphere:

Project	Location	Status
in the UK		
LCGrid	Portsmouth	LCGrid portal
RealityGrid	edinburgh	developing portal
GeneGrid	Belfast	developing portal
P-GRADE	westminster	portal for GEMLCA workflow
MyGrid	Manchester	developing portal from workbench tools
in Europe and Canada		
GridLab	Berlin	developers of GridSphere

5.2 CHEF

The **CompreHensive collaborativE Framework** (CHEF) project is developing a flexible environment for supporting distributed learning and collaborative work, and carrying out research.

This project is staffed by members of the University of Michigan School of Information, Media Union and Medical School. The developers are working closely with and are contributing to the OKI reference architecture from the Mellon Foundation and are collaborating with other groups interested in open source collaboration standards.

Target user communities include those involved in the scholarly activities of teaching, learning and research at the University of Michigan, and their students and colleagues involved in teaching, learning and research that are outside of the Michigan community.

CHEF is aimed at making available a set of functional elements that can be easily configured by users to accomplish a wide variety of activities. This framework supports existing and emerging capabilities, and seeks to make the integration of new functionality as easy as possible. It provides organization for the disparate functionality used to support research, collaborative and learning activities and combines locally developed, commercial off the shelf and free off the shelf components including various tools which have been found to be widely useful.

There is much commonality in the needs and tools used by people in the communities identified and the CHEF developers have seen the emergence of frameworks for user configurable toolset delivery. CHEF mobilizes that experience in the effort to develop a comprehensive framework to support these activities, and to make this framework available for wide use.

The following table shows some existing projects using CHEF:

Project	Location	Status
in the UK		
InfoPortal v3.0	Daresbury	migrating tools
e-HTPX	Daresbury	evaluating and developing portlets initially as JSPs
NGS	Daresbury	porting HPCPortal and OGCE tools
in the USA		
CMCS	PNNL	using with Ecce meta-data tools
BioGrid		using
Alliance Grid	Indiana	using with Xportlets
NPACI Grid	Texas	using with HotPoage GPIR
	U. Michigan	WTNG and CTNG tools migrating to Sakai
	MIT	tools migrating to Sakai
	Stanford	tools migrating to Sakai
	Indiana	tools migrating to Sakai

5.3 OGCE

In the USA several groups previously in the so-called Portal Collaboration 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 also referred to as the NMI portal. The Portal Collaboration principally involved the San Diego Supercomputer Centre with Perl-based GridPort and HotPage toolkits developed by Mary Thomas and Steve Mock supporting the NPACI Grid and NCSA at Urbana-Champaign with the Java-based GPDK toolkit developed by Jason Novotny. OGCE now includes researchers and end users from Indiana, NCSA, Michigan, Texas and Argonne National Lab. OGCE now embraces 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. Jason Novotny has gone on to develop GridSphere as a parallel activity with Mike Russell directly supporting the EU GridLab project as described above.

The functionality of OGCE includes a number of the collaboration and Grid tools mentioned below. A lot of this is explained in more detail in the paper by Gannon et al. [54].

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 noted in 2003 that it had 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 further 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 compose it into applications as components. This work may ultimately also find its way into the Sakai code base.

Portal deployments in the USA 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 Earthquake 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 was planned to be ported to Sakai by autumn 2004 (see note below from Marlon Pierce) so that it will be JSR-168 compliant. The generic OGCE release is available from <http://www.ogce.org>.

According to Marlon Pierce, lead developer on the OGCE project, 21/5/04, *In a nutshell, Sakai is an collaboration between educational middleware and portal developers. It's basic component projects are a) implementing a JSR-168 container (uPortal group leads), building education/ courseware and collaboration tools (UMich leads that, I think), and building OKI-compatible middleware services. OKI is a previously unimplemented standard for education services and objects from MIT.*

OGCE concentrates more on Grid/ science portal applications. We are experimenting with early Sakai portal releases as they become available, but they still have a ways to go. We are planning to use them primarily as a) our base JSR-168 container, and b) a way to resynch with the latest version of the CHEF tools. I don't have an updated timeline from Sakai, but we hope to have this as part of our next major portal release at Supercomputing [2004].

These comments partly drove our proposal to use Sakai as a VRE Demonstrator which received a supporting letter from Brad Wheeler, PI of the TeraGrid project.

The following table shows some existing projects using OGCE:

Project	Location	Status
in the UK		
in the USA		
NMI portal	Indiana Texas	using for US Grid projects
Alliance Grid		using with Xportlets
NPACI Grid		using with HotPage GPIR
LEAD		
TeraGrid		
DOE Fusion		
NeESGrid		

5.4 Sakai

Sakai is a \$6.8M project which implements a model for the purposeful coordinating of work in a large community of teachers and learners. It is based on many of the principles of open source development efforts, but community source efforts rely more explicitly on defined roles, responsibilities, and funded commitments by community members than some open source development models. The project is founded by the University of Michigan, Indiana University, MIT, Stanford, the uPortal Consortium, and the Open Knowledge Initiative (OKI) with the support of the Andrew W. Mellon Foundation and William Hewlett Foundation. The Sakai Educational Partners' Program (SEPP) extends this community source project to other academic institutions around the world as described in Section 3.

The consortium members are joining forces to integrate and synchronize their considerable quantity of educational software into a pre-integrated collection of open source tools. According to the developers, this will yield three big wins for sustainable economics and innovation in higher education:

- A framework that builds on the recently ratified JSR-168 portlet standard and the OKI open service interface definitions to create a services-based, enterprise portal for tool delivery. Since September 2004 this now includes WSRP;
- A re-factored set of educational software tools that blends the best of features from the participants' disparate software (e.g. course management systems, assessment tools, workflow, etc.);
- A synchronization of the progress of the SEPP members in developing and sharing a common set of open source software.

The stated products of this project will include an Enterprise Services-based Portal, a complete Course Management System with sophisticated assessment tools, a Research Support Collaboration System, a Workflow Engine, and a Technology Portability Profile as a clear standard for writing future tools that can extend this core set of educational applications.

Use of these modular, pre-integrated tools will in principle greatly reduce the implementation costs of one or more similar tools at any institution. The core partners are committing over \$2 million per year to launch and support this two year project. The core universities are also committed to implementing these tools at their own institutions starting in Fall 2004 through the duration of the project. The commitment of resources and adoption is purposefully set on an aggressive timeline to swiftly integrate and synchronize the educational software at the core institutions. This effort will demonstrate the compelling economics of "software code mobility" for higher education, and it will provide a clear roadmap for others to become part of an open source community.

Sakai 1.0 beta was released to SEPP on June 23. You can now download release candidate RC2 from <http://cvs.sakaiproject.org/release/1.0.rc2/>.

Technology Portability Profile

The technical barriers can be overcome by distilling the accumulated architectural knowledge and programming experience gained in building systems into a Technology Portability Profile (TPP) that provides four essential elements for code mobility.

- Service Interface Definitions – the Open Knowledge Initiative Open Services Interface Definitions (OSIDs) have provided an essential first contribution to solve this technical challenge. Local implementations of the OSIDs at an institution integrate heterogeneous local architectures (e.g. an authentication system or directory service) by using common connectors that enable code mobility for OKI-based application software (e.g. a CMS, library system, collaboration tool, etc.);
- Standard Portal – advanced course management systems (CMS) are based on portals that aggregate class information and services and allow the user to personalize and customize their views of these classes, services and information. At the same time, university-wide services are migrating from independent Web-based interfaces that accessed silo-like systems (e.g. Bursar, Library, Registrar, CMS, etc.) to enterprise-wide portals that integrate a personalized view of the full range of a university's services and information. The uPortal effort has brought forth a powerful portal environment that has commanded broad adoption, but it lacked the recently ratified JSR-168 portlet specification needed for tool interoperability. A standards-based portal that can be used as both the academic portal for the CMS as well as for delivering other university services via the JSR-168 portlet standard is a core building block of the TPP;
- Tool Interaction Framework – tools need a framework to provide a consistent way for invoking other tools and passing information among them. For example, a homework 'drop box' tool developed at one university may need to invoke and pass a grade to a 'grade book' tool developed at another. This framework provides a common place for these tools to interact with each other in a standard way. It also provides services like notification that cross tool boundaries. The U. Michigan CHEF project has developed such a portal-based framework which provides the environment for a large set of course management tools;
- Localized User Interface – institutional adoption by faculty and students often relies on user interfaces that match other familiar systems. A user interface includes colors, fonts, logos, and navigation aids that can be localized as needed without disturbing the underlying functionality of the software. The TPP will provide standard methods and description of best practices for Sakai-based tool interfaces to be customized and modified by user institutions

The maturing of the OKI OSIDs, recent demonstration of a working tool interoperability framework at U. Michigan, and industry ratification of the JSR-168 portlet specification made the timing perfect for developing the full Technology Portability Profile for higher education. But, while specifications, standards, and profiles are numerous in higher education, it is large adoptions that give a specification momentum to become a universal and wide spread standard. We noted that many of the requirements of the TPP are also appropriate to e-Research.

Sakai Java Technology

The Sakai core depends upon a number of novel technologies which are now described. The risks associated with these technologies are noted.

Sakai (<http://www.sakaiproject.org>) is basically a software package designed to add collaboration and course management facilities to the uPortal portal framework. The software already provides collaboration tools in the form of chat, discussion and shared file space, and is being extended with further tools designed to add course management functionality. Tools are being added for both test creation and assessment, the unspoken ambition being to create a functional open source competitor to Blackboard.

Look and Feel

Sakai takes the basic portal paradigm of portlets arranged in a tiled formation on screen and adds the ability to group users into 'worksites', seen as tabs across the top of the portal display. Sakai diverges somewhat from the concept of allowing a user to arrange several portlets on screen and then persisting that layout between user sessions. Instead, Sakai displays one portlet at a time and arranges a toolbar of buttons down the left hand side of the display. Whilst on the surface, the fact that you can no longer aggregate you own collection of portlets seems to be a waste of good portlet functionality, we think that having the portlets accessible quickly on a toolbar is a sound design decision as it simplifies usage. The single portlet per tool model only varies with the worksites Home tool. When you first click on a worksite tab, you are taken to the Home tool, which presents you with a synoptical view of the recent activity in the various worksite tools. You can view recent chat messages, postings in the dicussion tool and announcements on one screen.

Architecture

Sakai currently has a hybrid architecture aimed at allowing the gradual transition from CHEF style tools to Sakai style TPP tools. Like other portal projects Sakai uses the MVC, Model-View-Control, paradigm. Whereas CHEF uses Jetspeed as its portlet layout engine, Sakai uses uPortal. Whereas CHEF used the Apache project's Turbine as its component and persistence framework (Model), the Sakai TPP uses Spring. Whereas CHEF uses Velocity as its display (View), the Sakai TPP uses JSF (**J**ava **S**erver **F**aces). Whereas CHEF uses Struts as its intermediary (Controller) between the components and the display (View), the Sakai TPP uses JSF. The hybrid architecture arises from the requirement that CHEF tools also need to run alongside the TPP tools, so all the technologies mentioned in this section, bar Turbine, are present in the Sakai RC1 software stack. This hybrid approach enabled the basic CHEF collaboration tools to be brought straight across from CHEF with minimal modifications. This decision allowed the core Sakai team to work on architecture, rather than tool porting in the initial stages of the project. The downside of this is that the Sakai stack is large and more complex than it would be with a single coherent MVC framework in place.

Sakai is currently in a state of flux at release point v1.0 RC2. The chosen architecture for the software is being debated upon and there is a drive to prioritise the establishment of WSRP compliance above the establishment of JSR-168 (pluggable portlets) compliance. The Sakai core team are also working on uncoupling the software from particular portal frameworks and Java application servers (current uPortal and Tomcat respectively). Whilst these developments will ultimately be good for Sakai, the lack of API stability is causing problems with tool authors - a group who are expected to prefer API stability over elegance.

There will be some important differences between the v1.0 (15/9/04) and v2.0 (2Q05) releases.

The primary purpose of the v1.0 release is to run in production at the educational institutions of the core SEPP members and show others the general direction of the software. It will be production quality code which can be used, but it should not be considered the "final" Sakai. Within v1.0 the developers have a number of APIs that are only implemented as 'legacy' and all of the major tools are legacy, meaning "ported from CHEF".

Between the 1.0 and 2.0 release new Sakai framework capabilities are being developed and new tools built using those capabilities. There will also be a general tightening up of the environment for the development of new tools. By 2.0 the team hope to have a product that is on par with commercial collaborative systems in a number of important areas:

- Well defined and well documented development platform
- Ready-to install in a wide range of configurations out of the box
- Rich feature set

The key is that 1.0 may be lacking in some areas that is important for ease of porting new tools

As an example, 1.0 in production by default required either Oracle or MySQL, although the MySQL specific

code does not seem to be as well tested as is the case with Oracle. The reason for this is that the University of Michigan (the main developers of CHEF and Sakai) runs Oracle for all its student systems. The embedding of SQL code directly into the software is commonly regarded as bad practice; if some middleware like Hibernate or iBatis' sqlMaps were used it would be quite easy to port the mappings to various RDBMSs without touching a line of Java. It may well be that an abstraction approach has been tried and performed badly, hence the presence of tuned (RDBMS specific) code, but this is merely speculation. We have shown that we can adapt it to work with PostgreSQL by patching the various files involved, but this should be regarded as an interim measure. This is an example of the current minor 'annoyances' that an organization might encounter getting v1.0 into production.

Institutions choosing to implement v1.0 right now should therefore understand that they will need to have a little more developer talent allocated to Sakai than if they wait for v2.0. There may be something that you have to adjust locally in 1.0 that by 2.0 we will have as a simple option in install.

It is promised that Sakai v2.0 is just as easy to *install, configure, and forget* as commercial collaborative systems.

There is still currently a dialogue occurring, amongst the members of the SEPP and the core development team, regarding the architecture of Sakai. This dialogue was in part triggered by cross language support concerns raised at the first SEPP conference in Denver, see Section 3. What is now being proposed is a de-emphasising of JSR-168 compliance, with a corresponding emphasis on WSRP support. The reason for this is obvious – WSRP is language agnostic, so if Sakai can aggregate remote software services via the WSRP mechanism, it no longer can be accused of being Java specific. Another big shift in architecture is a proposed effort to uncouple Sakai from uPortal as its layout engine. There is now talk of being able to run Sakai with other portal frameworks like Jetspeed and Liferay. All of these shifts in architecture could be looked upon with concern, but what it highlights is the open nature of the development effort and the influence that the SEPP members now wield as a group. Sakai will work towards being more easily integrated into existing institutional infrastructures, with eased deployment and tool aggregation facilities, and that can only be a good thing. The downside to these changes is that early access tool developers like ourselves are faced with the frustration of a shifting architecture to code to.

The following table shows some existing projects using Sakai:

Project	Location	Status
in the UK		
ReDReSS	Lancaster	using v1.0
NCeSS	Manchester	evaluating
CARET	Cambridge	evaluating v1.5
in the USA		
	U. Michigan	WTNG and CTNG tools migrating to Sakai
	MIT	tools migrating to Sakai
	Stanford	tools migrating to Sakai
	Indiana	tools migrating to Sakai

5.4.1 uPortal

uPortal was used by Sakai up until version 1.0, see <http://www.uportal.org>. Indeed the uPortal Consortium is a member of the Sakai Collaboration. uPortal is an open-standard effort using Java, XML, JSP and J2EE. It is a collaborative development project with the effort shared among several of the JA-SIG member institutions. You may download uPortal and use it on your site at no cost.

The current version of uPortal supports only WSRP consumer interface. The next generation of uPortal (version 3.0) will be fully JSR-168 compliant and will support WSRP with both the producer and consumer

interfaces. The most significant change in uPortal v3.0 as compared to v2.3 is that native portlet support will be provided. Currently it is done using an adaptor into uPortal channels. Pluto will be used in both cases. This is further explained on the Web site.

5.4.2 JavaServer Faces GUI Layer

JavaServer Faces (JSF) technology [81, 74], is used to separate out the presentation from tool logic. JSF is a framework for building user interfaces for web applications and follows the Model-View-Controller (MVC) design pattern. JavaServer Faces technology includes:

- A set of APIs for: representing UI components and managing their state, handling events and input validation, defining page navigation, and supporting internationalization and accessibility;
- A JavaServer Pages (JSP) custom tag library for expressing a JavaServer Faces interface within a JSP page.

Designed to be flexible, JavaServer Faces technology leverages existing, standard UI and Web-tier concepts without limiting developers to a particular mark-up language, protocol, or client device. The UI component classes included with Java Server Faces technology encapsulate the component functionality, not the client-specific presentation, thus enabling JavaServer Faces UI components to be rendered to various client devices. By combining the UI component functionality with custom renderers, which define rendering attributes for a specific UI component, developers can construct custom tags to a particular client device. As a convenience, JavaServer Faces technology provides a custom renderer and a JSP custom tag library for rendering to an HTML client, allowing developers of Java 2 Platform, Enterprise Edition (J2EE) applications to use JavaServer Faces technology in their applications.

Ease-of-use being the primary goal, the JavaServer Faces architecture clearly defines a separation between application logic and presentation while making it easy to connect the presentation layer to the application code. This design enables each member of a Web application development team to focus on his or her piece of the development process, and it also provides a simple programming model to link the pieces together. For example, Web page developers with no programming expertise can use JavaServer Faces UI component tags to link to application code from within a Web page without writing any scripts. Unfortunately, JSF uses a set of custom XML based tags for rendering the interface, so web developers who are used to using WYSIWYG tools for HTML development are going to have to change their working pattern until appropriate tooling appears. Another problem JSF may face with logic programmers will be a sense of ambivalence: why use JSF when Struts/Tapestry/WebWork work so well? Sakai tools can be written using web frameworks other than JSF, although it will be more tricky to integrate with the Spring-based Sakai component repository. JSF and Spring are core parts of the TPP for this reason.

Developed through the Java Community Process under JSR-127, JavaServer Faces technology establishes the standard for building server-side user interfaces. With the contributions of the expert group, the JavaServer Faces APIs are being designed so that they can be leveraged by tools that will make web application development even easier. Several respected tools vendors are contributing members to the JSR-127 expert group and are committed to supporting the JavaServer Faces technology in their tools, thus promoting the adoption of the JavaServer Faces technology standard.

The expert group is actively developing the specification for JavaServer Faces. To check its current status, see the JSR-127.

5.4.3 JSR-168

JSR-168 is the Portlet standard ratified by JA-SIG in August 2003 following community comment. An open source implementation is available, Pluto is from the Jetspeed developers. See Section 1.2.

5.4.4 WSRP

WSRP: **W**eb **S**ervices for **R**emote **P**ortlets is a Web services standard ratified by OASIS in August 2003. See Section 1.2.

5.4.5 OKI Open Service Interface Definitions

O.K.I., the Open Knowledge Initiative, develops specifications that describe how the components of an educational software environment communicate with each other and with other enterprise systems. O.K.I. specifications address broad interoperability agreements that allow for adaptation and further specification by communities of practice. In this way, O.K.I. seeks to open new markets for educational tools and content.

The O.K.I. team at MIT continues to provide architectural and technical leadership for software initiatives in higher education, among industry affiliates, and through national and international organizations such as the IMS Global Learning Consortium. This work helps ensure that the next generation of e-learning software will integrate seamlessly with the educational enterprise, and it establishes O.K.I. as a global leader in behavioural specifications for educational technology interoperability.

The Open Knowledge Initiative was initially funded through a grant from the Andrew W. Mellon foundation. This work is now being taken forward in the Sakai project through the design of a pragmatic portlet API to the OSIDs, Open Standard Interface Definitions. The Java specification for the OSIDs is available from SourceForge at <http://sourceforge.net/projects/okiproject/>.

The Open Knowledge Initiative (OKI) is a multi-institutional, multi-year project led by MIT and Stanford and funded by the Andrew W. Mellon Foundation. It is a highly collaborative effort to enhance learning by creating an open programming environment that supports sharing and pedagogical experimentation. The initiative is in its second year of work, and has spawned much interest and high expectations in the international educational community. It is important that expectations for the products of the OKI are realistic, especially in light of rapid growth in enterprise-wide academic systems such as Blackboard and WebCT, the academic counterparts to our administrative systems.

The following summary of the initiative is couched in non-technical terms to aid small colleges and universities as they prepare to use and to contribute to the OKI.

The OKI has short- and long-term goals. The short term goals are to set standards for software development, to develop a set of basic software tools which are needed by most educational software, and finally to create some exemplary educational software. Many of these short-term goals have been met. The long-term goals are to create a community of software developers who can adopt, extend and improve on the OKI standards and software, and to ensure that the Initiative can be sustained in the face of constantly changing technologies.

At the core of the OKI are fully public software standards that define an open programming environment to foster pedagogical experimentation and sharing. This open programming environment is based on delivery of learning materials through standard web browsers. This means that software written to the OKI standards will be accessible to users of nearly all modern computers, and the content will be accessible wherever there is an Internet connection.

The OKI can be most easily understood when viewed as a set of four layers. Each activity of the Initiative is based upon the layer below it. The bottom-most layer is the infrastructure of the campus: the campus network and associated hardware and software that is the responsibility of each campus. Built upon this foundational layer are OKI "common services" and "educational services." Common services are modules of software that perform relatively limited functions. Educational services comprise the next layer and perform more complex functions. Many educational services are created by the integration of several common services. The OKI defines the function of these services and how they interact with one another. Finally, at the topmost layer are "educational applications" which are complex systems that integrate common services and educational services into more comprehensive learning systems.

The availability of publicly defined common services and educational services makes it possible for programmers to focus on their pedagogical goals by using and reusing software modules as building blocks for more complex educational software. For instance, the OKI defines common and educational services such as how students and faculty log into an OKI-compliant system, how workgroups of learners are managed, or how files are shared among students and faculty. Programmers will need many such services to integrate into more complex educational software that meet pedagogical needs.

An educational application is software that has one or more specific pedagogical goals and is built upon the architecture defined by the OKI. For instance, student electronic portfolios could be an educational application. An e-portfolio application might integrate many OKI-defined services such as authenticating the identity of a student, managing the files in a portfolio, and allowing portfolio files to be annotated and assessed.

OKI staff members have not only defined the standards for these services, they are implementing many of them so that developers of educational software can take early advantage of the OKI effort. The common services are being written in the Java programming language but because the definitions of these programs are public, others can create the same functionality by writing their own services in programming languages of their choice. Thus, the OKI is not dependent upon a single programming language or a single computing platform for shared development or access to educational content.

It is important to note that the OKI is not a comprehensive course management system, although comprehensive educational applications are being developed on the OKI standards. These included <http://aboutcoursework.stanford.edu/> Stanford's CourseWork, <http://stellar.mit.edu/> MIT's Stellar and <http://www.chefproject.org> The University of Michigan's CHEF among others. For instance, the latest release of CourseWork complies with many of the OKI standards and is being assessed in courses at several institutions this academic year.

It is also important to mention that the Initiative will not produce 'free' software. The OKI is a non-commercial project based on the 'open-source' model of software licensing. This gives us the freedom to adapt the products of OKI to our needs, but supporting our use of OKI products will not be free. Our institutions will not avoid the costs of course management systems simply by adopting the products of the OKI.

Open source software is distributed freely and includes all the source materials that will allow a programmer to improve the software, adapt it to their needs and to fix bugs. Open source software is thus maintained and extended by a community of programmers. However, the usual avenues for software support don't exist for open source software that is distributed freely. There is not likely to be a 24-hour phone number to call for help. More likely, our institutions' knowledgeable support staff will be able to post inquiries to an e-mail list or browse web resources for answers to support questions. Thus technical support for open source software is a cost center.

Our institutions will be able to adopt the products of the OKI if they have the staff time and expertise to install and maintain the software they select. The costs of using OKI products will come from meeting hardware needs for the products (servers, backup, etc), installing the applications, connecting them to institutional databases as needed, and supporting the local user community. The benefits of the OKI

will come as we gain access to diverse educational applications contributed to the OKI community by colleges, universities and other developers. For example, Stanford University is developing an assignment and assessment manager to deliver online assessments that are discipline-specific and teaching method-specific, and to allow faculty to embed assessments into online content for better formative feedback.

Commercial developers are able to contribute educational applications to the OKI (and Blackboard and WebCT have announced they will), but we will have to license those commercial products even if they do fit into the freely distributed OKI framework.

Our institutions will be able to contribute to the OKI if they have the staff time and expertise to develop new modules that comply with the OKI standards. Developing OKI applications implies a higher level of resource commitment and expertise than just using the products of the OKI. Several major institutions in the US and UK are core collaborators or application developers for the OKI, so the outlook for a rich suite of educational applications is excellent. There is still plenty of room for the smaller colleges and universities to contribute. For many of our institutions, a strong focus on undergraduate learning outcomes puts us in a unique position to design effective educational applications in the OKI framework. Our ability to implement such applications will depend on our commitment of technical resources and the challenges of integrating basic OKI services into effective pedagogical software. Those who wish to contribute to the OKI effort should participate in the OKI Developer's Network the first meeting of which will take place at MIT in March, 2003.

For much more official information about the Open Knowledge Initiative, visit <http://web.mit.edu/oki/>.

The Open Knowledge Initiative has produced a series of Open Service Interface Definitions (OSIDs) informed by a broad architectural view of the educational technology landscape. As learning management systems have become a core component of the campus information technology infrastructure, O.K.I. seeks to simplify and enhance the creation of educational applications.

The OSIDs are an abstraction layer between the programmer and the enterprise infrastructure systems of his or her campus. Each OSID is characterized by a tightly defined set of methods and strict boundaries. This approach offers a number of important benefits to applications designed to the OSIDs:

- Simple integration with existing infrastructure;
- Local innovations can be shared across campuses or universities;
- Adaptation to new technology without destabilizing the overall environment;

The OSIDs are divided into the following classes. See also <http://www.grid.ac.uk/ETF/public/WebServices/classes.html>. We note that many of these are not mature and the descriptions are sketchy.

Common services:

Agent: no information given

Authentication: The Authentication OSID gathers required credentials from an agent, vouches for their authenticity and introduces the agent to the system.

Authorization: The Authorization OSID allows an application to establish and query a user's privileges to view, create, or modify application data, or use application functionality.

Dictionary: The Dictionary OSID provides a means to support multiple languages, domain-specific nomenclature and culture-specific conventions through interchangeable property files.

Filing: The Filing OSID provides platform-independent means to handle files arranged in simple hierarchical containers.

Hierarchy: The Hierarchy OSID manages parent-child relationships among elements. In addition to simple tree structures, the OSID supports hierarchy that are recursive and have nodes with multiple parents.

ID: no information given

Logging: The Logging OSID records and retrieves a variety of application activity history

Scheduling: The Scheduling OSID manages events in shared calendars.

Shared: The Shared OSID contains fundamental objects used in the other OSIDs to provide their functionality.

SQL: The SQL OSID provides relational database access functionality at a higher level of abstraction than the DBC OSID. Unlike DBC, it is not dependent on JDBC.

User Messaging: The User messaging OSID supports communication and notification among users.

Workflow: The Workflow OSID provides a way to manage an interdependent succession of activities each of which has completion constraints.

Educational services:

Assessment: OKI Assessment OSID provides APIs for managing banks of items, sections, and assessments, and for publishing assessments.

Course Management: A Course management service allows applications or services to access and manage courses, modules and other units of learning.

Grading: Part of the Assessment process. A Grading service supports submitting grades against courses, modules, and other units of learning.

Repository: A Resource management service supports the management of finite physical resources, such as equipment and rooms.

Java and PHP implementations are available from SourceForge.

5.4.6 Spring

Spring is an in-memory component repository where developers can isolate common functionality for use in multiple areas of an application. The core paradigm behind Spring is the Inversion of Control (IoC) pattern where references to the common component are injected into other components (Java beans for example) that have a need of the functionality implemented. The consumer is only ever aware that it needs an implementation of a particular API, it doesn't care how that is implemented (an example of the Strategy Pattern). Sakai uses Spring extensively to factor out functionality common to the various Sakai services.

5.4.7 Database and Hibernate

The Sakai code comes preconfigured to use Hypersonic SQL (HSQL), an in-memory relational database engine. It can also be configured, at deployment time, to use either Oracle or MySQL. The object/relational mapping (ORM) software Hibernate is made available by the Sakai framework to allow TPP tools to effectively define their own database tables and automatically forward generate them. Sakai still, at this point, uses SQL statements directly embedded in the Java source code, in many cases this SQL uses Oracle or MySQL specific syntax. In fact, Sakai releases seem to be tilted more toward Oracle than any other RDBMS, this is probably due to the fact that the University of Michigan uses Oracle for most of its database needs. This is an area of concern if you wish to use another RDBMS like PostgreSQL. Ideally Sakai should be using a Hibernate (or perhaps iBatis' sqlMaps) layer for ALL of its database requirements.

5.5 Available Portlet Tools

We here follow the classification scheme of Appendix H and identify what portlets are available from various sources which are or could be incorporated in the Sakai or other JSR-168 compliant framework. The evaluation tasks reported in the other appendices have showed in some cases how this can be done.

5.5.1 Collaboration

Some collaboration portlets available include:

- Login/ Logout - as it says on the tin
- Membership - join/ unjoin from workspaces
- Schedule - calendar for workspace
- Resources - resource list and links
- News - news service
- Discussion - threaded discussion service
- Chat - online chat room
- Anabas - join and participate in Anabas Impromptu collaborative session via applets
- Newsgroup (Read/ post) - topic-based threaded newsgroup tool using Indiana XML schema
- Newsgroup (Request) - request a new topic
- Newsgroup (Admin) - for admins only!
- Bibtex (Read/ edit) - add/ edit entries in Bibtex topics for shared authorship of reference lists. Uses Indiana XML schema
- Bibtex (Request) - request new Bibtex topic
- Bibtex (Admin) - for admins only!
- Bibtex (Super Admin) - for super admins only!
- Customize - chose which tools you want in the workspace, edit portlet layouts etc.

5.5.2 e-Research

Some e-Research portlets, mainly for managing remote jobs on a computational Grid include:

- Proxy Manager - store and retrieve credentials from MyProxy repository
- GRAM Job Launcher - launch a Globus job via interactive GRAM protocol
- GridPing - test a remote Grid service
- Grid Job Submission - submit a remote batch job to a Globus resource
- LDAP Browser - browse Globus MDS Grid Information Service for remote resources
- GridFTP - invoke 3rd party file transfer
- GridContext - browse and manage DNs in Xdirectory server - provides context services for managing metadata
- GPIR Browser - browse Grid resources via Grid Port Information Repository giving information on HPC status, jobs, etc.
- Historical GPIR - use GPIR to view stored information
- CSF Job Submission - submit job to a Community Scheduling Framework service
- Sequencer - ditto via GridPort sequencing service
- OGRELancher - Open Grid Computing Runtime Engine: launch simulation event via OGRE from NCSA, workflow based on ANT or BPEL
- OGREEventView - monitor OGRE events
- COG-Workflow - use workflow enactment via the Java CoG kit
- App Manager - register, de-register and edit information about a Grid application
- Condor - submit jobs via Condor or DAGMan and monitor them

- Customize - chose which tools you want in the workspace, edit portlet layouts etc.

5.5.3 e-Learning

Sakai comes pre-bundled with various tools applicable to an e-Learning context. There is a threaded discussion tool, chat room tool and file sharing tools, including a tool designed to allow students to upload files for the worksite administrator's eyes only (the drop box tool). Most of Sakai's out of the box functionality is, in fact, geared towards e-Learning, as is the user interface model of worksites and tools.

5.5.4 Digital Information

We have no information about digital information services in Sakai. Services from the JISC IE programme, such as cross search services, could be included.

5.6 Outcome of Evaluation Work Packages

This section reports on the outcome of the investigations and tests carried out in this Sakai Evaluation Exercise.

5.6.1 Evaluation Report Part 1: Technology Survey

The Technology evaluation Report draws upon two sources. Firstly the proceedings of the *Portals and Portlets 2003* workshop [28] which was written up in early 2004. Secondly the technology survey undertaken by Adrian Fish for the ReDRess project [37]. The body of this report is provided in Appendix A.

5.6.2 A Review of the Issues for building standards-compliant Portlets

Sakai does not as yet support the JSR 168 portlet standard and is, at version 1.5 starting to add WSRP support. The TPP can be used as the template for constructing and subsequently sharing Sakai tools. A wise strategy at this point would be to code the minimum of tool logic in the Sakai TPP specific part of the tool and factor out the main logic into delegates, ready for a recoding of the tools into the JSR 168 standard. On the other hand, the TPP offers up a lot more functionality from the framework than seems to be the case with JSR 168. You can use Hibernate to create a database structure required by your tool, for example.

5.6.3 An Assessment of the Potential of Sakai as a Platform for Customised Portals

Sakai is a useful software framework for developing customised portals. Once you are familiar with the components of the TPP, mainly Spring and JSF you can create and deploy portlet tools in a couple of hours. Development follows the standard web application pattern very closely; in fact every tool is a web application that the Sakai dispatcher is made aware of via a registration XML file. Adrian Fish has written tools to make Grid MDS queries via a web service at Daresbury and to query metadata repositories, again via a web service at Lancaster. The main problem with tool deployment at the moment is that only a user with global admin status on Sakai can add the tool to the worksite, as opposed to each worksite's nominated administrators. If we disregard the inflexibility of the worksite admin model, Sakai fulfils the requirements for a useful software framework and deployment platform.

5.6.4 Evaluation Report Part 2: Developer and User Feedback

This report focuses on issues of ease of administration, user management, customisation, etc. The body of this report is provided in Appendix B.

During the time frame of this evaluation exercise, we have had dialogues with a large number of people engaged in state of the art research in e-Science projects funded via the Research Councils, Core Programme or DTL. Many have been supportive to the idea of using portal interfaces and of developing a VRE encapsulating wide-ranging functionality. In fact there is no doubt that a VRE building on, and extending the capabilities of the current Grid and using Web services is a high-priority goal for many areas of applied research. Collaboration tools are high on the list of desirables. Portals are one means of delivering the functionality of this VRE, but additional requirements for lightweight toolkits have also been expressed by several groups [60, 58, 59, 62].

5.6.5 Software Template for Sakai Institutional Adapters

See the Sakai Evaluation worksite at <http://e-science.lancs.ac.uk:8080/portal>. You must apply for a Sakai login if you do not already have one. Send an email to Adrian Fish at a.fish@lancaster.ac.uk with 'Re: Sakai Account Request' as the subject. You can then log in and join the Sakai Evaluation worksite via the 'Membership' tool visible on the left of your Web browser.

5.6.6 Software Template for Grid Tool Wrappers for use in Sakai/CHEF

See above

5.6.7 Roadmap for a UK Virtual Research Environment

A *Roadmap for a UK Virtual Research Environment* was prepared for the JISC Joint Committee for the Support of Research with the VRE Working Group chaired by Dr. David Boyd in April-May 2004 [45]. Along with this report a comprehensive service classification was produced with input from the Grid Engineering Task Force, JISC Information Environment and JCLT architecture groups. See <http://www.grid.ac.uk/ETF/public/WebServices/classes.html>.

5.7 Our Proposed Development Strategy

Sakai is currently the aggregation tool of choice for the ReDReSS project (<https://e-science.lancs.ac.uk/redress>) and the current instability is causing problems with tool development at the present time. What is proposed is a spreading of risk during the further development of the ReDReSS portal and Sakai VRE Demonstrator, using the following steps:

1. Continue to run a stable version of Sakai RC1 or RC2 on the ReDReSS Web site. This will continue to supply collaboration facilities and a reasonably structured place to hold project material. The instructions for installing such a Sakai instance are contained in a separate document [73];
2. Install GridSphere (<http://www.gridisphere.org>) with the aim of using this installation to test the JSR-168 compliance of any tools the projects produce;

3. Write any tools in such a fashion that both JSR-168 and Sakai TPP versions are generated at compile time. The TPP versions get tested in an isolated Sakai instance (currently Adrian Fish's workstation) before live deployment. The JSR-168 version gets deployed into GridSphere and thus tested for compliance;
4. When Sakai reaches the point of architectural stability, any tools deployed in GridSphere are moved to the latest Sakai version running on the Lancaster CeS Web server;
5. A similar procedure is adopted at Daresbury on a separate Web server.

6 Technology Evaluation Report and Conclusions

6.1 Introduction

This work package compared Sakai/ CHEF with Alternative Frameworks for a VRE. By a "framework" we really imply a means to deliver the VRE services in an appropriate way to end users and developers. This comparison is restricted to the high-level open source and platform independent frameworks, that is, Sakai, Gridsphere, CHEF/ OGCE, and Bodington. For the purposes of this report, we regard uPortal and Jetspeed as enabling technologies (portal containers) and do not review them here, indeed there are many others which fall into this class. This comparison is an ongoing task and lasted for the duration of the project as much of the technology and functionality is still changing.

6.2 Method

We have chosen to adopt a Multiattribute Utility Evaluation approach [87] to evaluating the alternative contenders for a VRE. This approach is nevertheless straightforward and simple to apply. The core of the procedure is to identify the most relevant values or criteria that are appropriate to the functioning of a VRE. Measurements are then made to determine the degree to which the criteria are attained. By doing so systematically, and by making numerical judgements wherever possible, we can compare the VRE contenders on a more objective basis than is usually the case. We have identified the following 10 broad criteria, some of which are further subdivided.

These criteria are as follows:

Criterion A: Can the framework be applied to all disciplines?

Criterion B: Is the VRE framework useful for users from both the e-science (a) and e-learning (b) domains?

Criterion C: Criterion C is divided up into two sub criteria:

a) It is generally accepted that software modules that are used by many projects end up being robust and well understood due to the amount of exposure they receive. It thus makes good sense to make use of publicly available libraries when building a software product as opposed to writing the same algorithms over and over again. This criterion is thus intended to reflect the degree of use of open source libraries by the VRE.

b) Conformance to ratified standards is another feature that is generally seen as being important. Standards conformance fosters ease of interoperability and ease of extension via plug-able components. JSR 168 and WSRP have been identified as the main two standards that, when adhered to, will allow Java components and Web service based components to be added to a VRE. What interface standardization also facilitates is reuse. There are many tools currently in circulation, written in Java, which could be re-factored to allow them to be plugged into a standards-based interface like JSR 168.

There are also many tools written in other languages. As long as these tools can be re-factored to talk the WSRP protocol, then they could also be re-used in a WSRP compliant container.

Criterion D: Make UK services and resources available in familiar environments e.g. typically via a Web browser;

Criterion E: Any open source project worth its salt is supported by a decent online community. Discussion forums, chat rooms and mailing lists have all proven to be incredibly useful tools for spreading know how about a software product. The following two criteria are an attempt to measure this kind of support.

a) This measures the degree of support there is for developers who wish to start writing tools for, or even extending, the VRE framework.

b) This measures the degree of support there is for users and administrators who wish to install, configure and use the VRE framework.

Criterion F: Offer choice in presentation or delivery for (a) services and (b) tools;

Criterion G: How steep is the learning curve required to use the VRE framework?

Criterion H: This criterion measures the amount of functionality you get “out of the box” from a particular framework. In other words, if an institution installed the basic version of this VRE framework and just left it at that, would it be any use?

Criterion I: Presence and extent of a future funding stream, for (a) <12 , (b) >12 and <24 and (c) >24 months

Have all the criteria been listed? There are others we could include like the track record of the developer team, but this is taken into account when we allocate a score under sub criterion F(a). We felt that other, perhaps domain specific criteria, would be less important in an overall evaluation and are thus likely to have a lower impact and not affect the overall rankings. This has been partially tested with a sensitivity analysis (see below).

We then ranked the 9 criteria in order of importance and allocated a score out of 10, this was then standardized to sum to 1. If criteria had a lot of overlap with another criterion they would be given similar rankings. For simplicity, and to start with, we have given them all equal weights (0.1).

The same process was then applied to the sub criteria. This gives us the weights w_{X_j} that can be used to aggregate the score of each component of criterion X to produce a total score for criterion X . So for example, for criterion B which is subdivided into 4 components, the total score for criterion B is given by $B = \sum_j w_{X_j} B_j$. The total score over all the criteria is then given by $U = \sum_X r_X X$, where r_X is the weight applied to criterion X . The weightings we used are shown in parentheses in the following figure.

The scores (out of 100) for each component were then obtained using our judgment.

6.3 Score Justification

Criterion A

All the platforms tested are domain agnostic before specialisation; it is the tools that make them domain specific. They all score 50.

Criterion B

Sakai: a) In its current form Sakai is primarily an e-Collaboration framework, in that there are facilities for online collaboration pre-bundled. There is also currently a software patch that allows Sakai to pick up

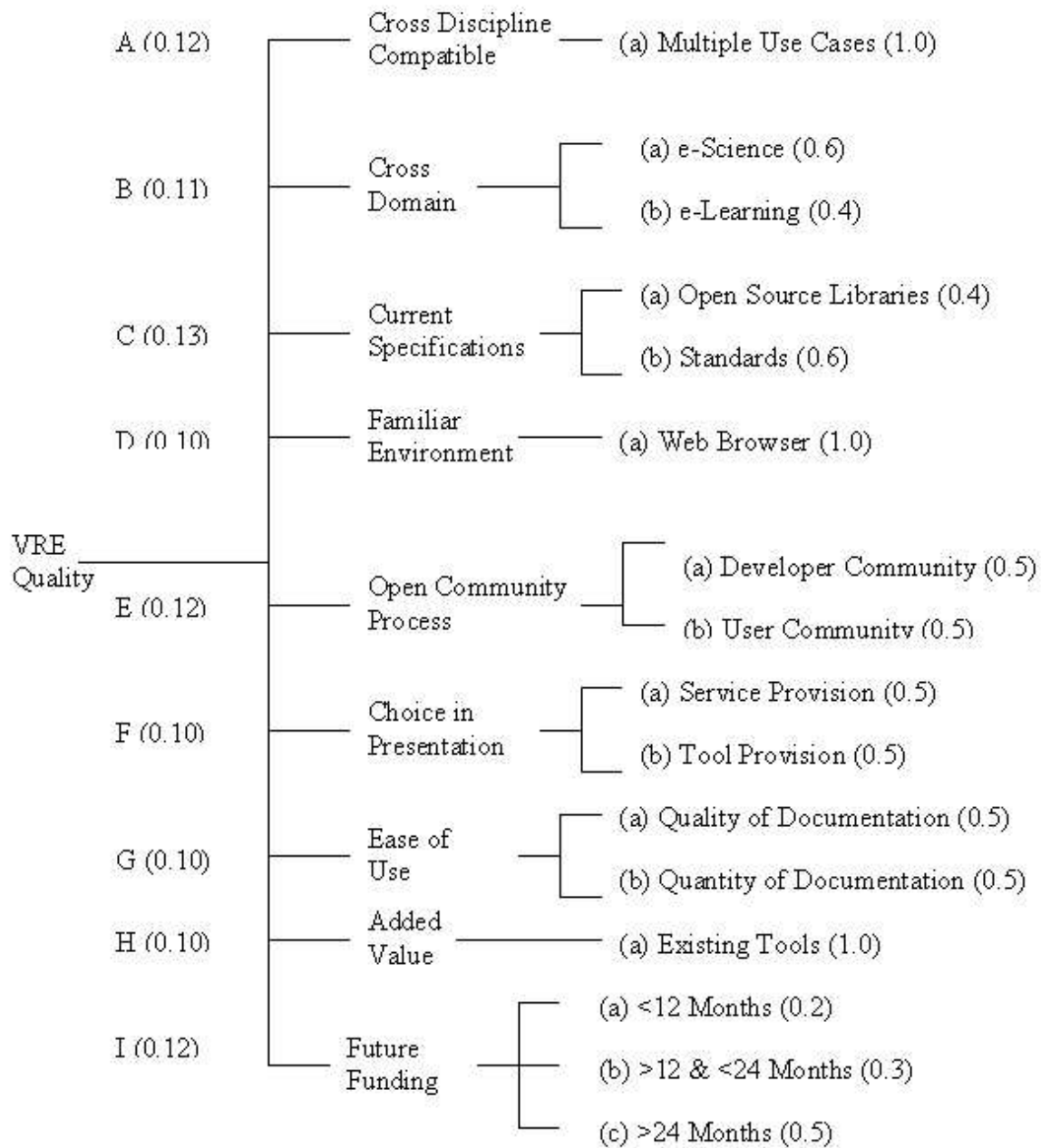


Figure 6: Value tree for comparing open source, platform independent VREs

X.509 certificates from MyProxy servers for Grid computing use. Sakai thus scores 60 for the e-Science sub criterion.

b) Sakai can also be used for e-Learning, as CHEF is at U. Michigan. It doesn't yet offer any specialized tools for things like assessment and IMS package delivery, but it does offer a working paradigm that lends itself well to representing courses and the students on those courses. Tutorial sessions can be held online, using the bundled collaboration tools, and course materials can be presented in the form of the multiple web pages that you can assign to each course worksite. Sakai scores 50 in the e-Learning category.

CHEF/ OGCE: a) The OGCE framework, based on a populated version of CHEF, is the clear winner in the e-Science category. OGCE makes use of CHEFS e-collaboration tools and adds grid computing tool components. NEESGrid in the USA use it in their earthquake simulation experiments. OGCE scores 70 in the e-Science category.

b) CHEF/ OGCE scores slightly less than Sakai in the e-Learning category as it offers similar facilities. See the Sakai section for more details. CHEF/ OGCE scores 45 here.

Bodington: a) Bodington is primarily of use as a learning environment. It has strong content management facilities and has tools for generating and displaying assessments complying with IMS QTI (Question and Test Interchange) format. It is, however, not of as great use to the e-Science community, who need collaboration tools and user grouping and administration features more than content navigation. Bodington scores 30 here.

b) This is where Bodington is more fitting. Bodington is a content manager by design, it uses a library metaphor to arrange content into floors and rooms; this is obviously a successful approach as Bodington has been in production at Leeds for around 4 years. A drawback of the Bodington approach is in the centralization of content as opposed to a strategy of organizing metadata on remote content. Bodington scores 70 here.

GridSphere: a) GridSphere is a portal framework in the sense that it is an unspecialized container for components that, when added will increase its utility within a specified domain. The main take-up of GridSphere has been within the hard science community and several projects have successfully added portlet components to GridSphere and put it into production. At the GridSphere website you can download a collection of Grid portlets for accessing Globus functionality. GridSphere scores 70 here due to its good Globus Grid tool support.

b) GridSphere scores poorly in the e-Learning sub criterion. There are no tools bundled, or in the pipeline, that are applicable to this domain. GridSphere is a framework designed for specialization however, so the potential is there to add e-portfolio, course management and assessment tools. GridSphere scores 20.

Criterion C

Sakai: a) Sakai makes good use of open source libraries from other projects. It uses the Spring framework, Hibernate, Java Server Faces, Pluto and Velocity amongst many others. It is also re-using lots of the CHEF tool code, which has been tested in the live environment of U. Michigan. For these reasons Sakai scores well and gets 70.

b) Sakai uses the OKI OSIDs for component messaging internally, although these interfaces are not exposed to external tools. There is no support for WSRP or JSR 168 tools presently, JSR 168 support has been in the pipeline for some time and WSRP is being worked into the architecture design at the time of writing. In reality, the work on formalizing the software interfaces used in Sakai is still ongoing

CHEF/ OGCE: a) CHEF uses open source libraries from other projects. It uses Struts, Velocity. JUnit and James, amongst others. CHEF scores 60. On b) CHEF scores 20.

Bodington: a) Bodington is a Web application written using servlets for its dynamic aspects. Bodington

does not leverage any Web application development frameworks like Java Server Faces, Struts or Tapestry, primarily for the reason that the architecture was effectively formulated in 1998, before the advent of these frameworks. Bodington also doesn't make use of any software libraries that aren't included in the Tomcat installation. As this metric depends on the utilization of cross-project libraries, Bodington scores 30.

b) Bodington's modularity comes at the Java Servlet level. If you want a new tool, you write a servlet. Data repository (DR) services are provided, but are not, to the authors' knowledge, currently based on ratified standards. As with all the other frameworks discussed, you cannot aggregate either JSR 168 or WSRP compliant portlets, so tool creation has to be done by proprietary means. Bodington scores 20 here.

GridSphere: a) Gridsphere scores 80. On b) GridSphere scores 50.

Criterion D

All the frameworks considered score 100 here as they can be displayed in any CSS and JavaScript enabled Web browser.

Criterion E

Sakai: The Sakai project has a formalized community process in place called the SEPP (Sakai Educational Partners Program). The need for the SEPP was actually expressed by the main Sakai funders, the Mellon Foundation, at the Sakai project's inception, and as such is a pre-requisite for funding to continue. To join the SEPP, institutions have to pay a fee of \$10K per year for three years. Now that version 1 has been publicly released, there are several public mailing lists for developers and users outside the SEPP. The mailing lists have been dedicated slight more to developers, so Sakai scores 70 and 60 respectively.

CHEF/ OGCE: The OGCE project has a good selection of majordomo mailing lists, and a large contact list on its Web site. OGCE scores 60 in each category.

Bodington: Bodington is a SourceForge hosted project and has a mailing list dedicated to developers. Traffic is relatively high and this reflects the list's utility. It would be useful if the lists were broken down into more specialized subject areas as there is a lot of material on them. Bodington scores 60 and 30 respectively.

GridSphere: GridSphere has both a user's and developers mailing list and so scores 60 in each criterion.

Criterion F

All the platform tested are presentation agnostic, so they all score 50.

Criterion G

Sakai: Sakai's documentation is patchy and still under development. Seeing as the architecture is still in flux, this is understandable. The state of the documentation is made up for to a degree by the quality of Sakai's community efforts. Sakai scores 20 in each sub criterion.

CHEF/ OGCE: OGCE's documentation is better, and more copious than Sakai's, as befits a longer running project. The fact that CHEF long since standardised on its architecture obviously helps. CHEF gets 50 in each category.

GridSphere: GridSphere's documentation scores 60 for both quality and quantity; there is a collection of documents in the form of "howtos", both on the Web site and in the binary distribution.

Bodington: Bodington scores 40 in each sub criterion. The Bodington gatehouse site hosts a collection of tutorials for students wishing to use the system but these are obviously targeted to its use as a VLE not a VRE.

Criterion H

CHEF/ OGCE is the clear winner here as it comes with all the collaboration tools of Sakai, plus a suite of useful Grid tools. Sakai comes second, as it comes supplied with a set of useful collaboration tools such as chat, threaded discussion and shared file space. GridSphere comes third as it comes with chat, file space and chart creation tools. Bodington comes last in this criterion, due to the fact that the out of the box toolset relevant to a VRE is sparse, as its intended role is that of an e-Learning/ content management environment, not a tool delivery framework.

Criterion I

Sakai: Sakai funding has only just got going, so we have given it 60 for each sub-criterion.

CHEF/ OGCE: CHEF is effectively undergoing a transformation into Sakai and it is unclear what the funding situation currently is. CHEF is still in production use at U. Michigan. CHEF gets 50,10,0 in each sub criterion respectively.

Bodington: Bodington is funded by individual project grants from funding bodies, it does not have a concrete funding strategy like Sakai. There is however considerable institutional commitment in the UK and this should really be considered in a funding assessment. As far as we are aware there are no Bodington projects funded past two years scores, so it scores 50, 50 and 20 respectively.

GridSphere: GridSphere funding is about to end. Furthermore, GridSphere is currently trying to get funding from Deutsche-Grid initiative to continue support for GridSphere, but now it looks like it may be only enough money to support only 1 developer for 3 years in the worst case, so we have only given them 20 in each sub-criterion.

6.4 Combining the scores and weights

																		Total	Ranks
Criteria	A	B		C		D	E		F		G		H	I					
Criteria weights	0.12	0.11		0.13		0.1	0.12		0.1		0.1		0.1	0.12				1	
Sub Criteria	a	a	b	a	b	a	a	b	a	b	a	b	a	a	b	c			
Sub Weights	1	0.6	0.4	0.4	0.6	1	0.5	0.5	0.5	0.5	0.5	0.5	1	0.2	0.3	0.5			
Sakai 1	50	60	50	70	30	100	70	60	20	30	20	20	60	60	60	60	53.64	1st	
Chef/OGCE	50	70	45	60	20	100	60	60	20	30	50	50	70	50	10	0	50.54	2nd	
Bodington	50	30	70	30	20	100	60	30	20	30	40	40	30	50	50	20	43.28	4th	
GridSphere	50	70	20	80	50	100	40	50	20	30	60	60	40	20	20	20	49.86	3rd	

Figure 7: Combining Scores and Weights

Results

We also performed a sensitivity analysis (1) by changing the weights, (2) having someone else decide on their own scores, (3) by dropping the least important criteria (D,F,G and H). In each situation we obtain very similar ranks, Sakai always comes out on top.

7 Acknowledgements

We acknowledge the following people who have freely provided their comments and input to the work reported here.

Alison Allden, Chris Awre, Mark Baker, David Boyd, David de Roure, Geoffrey Fox, Dennis Gannon, Glen Golden, Liz Lyon, Mark Norton, Jason Novotny, Marlon Pierce, Andy Powell, Alan Robiette, Charles Severance, Mary Thomas, Brad Wheeler, Scott Wilson,

We especially thank Mary Thomas, Dennis Gannon, Jason Novotny, Charles Severance, Glen Golden, Brad Wheeler and Jim Farmer for their continual guidance and help.

References

- [1] R.J. Allan *Survey of Computational Grid, Meta-computing and Network-based Tools* Edition 2 Technical Report DL-TR-99-02 (Daresbury Laboratory, February 2000) <http://esc.dl.ac.uk/StarterKit>
- [2] R.J. Allan *Resource Selection and Billing via Agents, Brokering and a Grid Hub* (CLRC e-Science Centre, 2000) draft
- [3] R.J. Allan, D. Hanlon, R.F. Fowler and C. Greenough *A Globus Developers' Guide with Installation and Maintenance Hints* Draft Technical Report, UKHEC Collaboration and CLRC e-Science Centre (2001) <http://www.dl.ac.uk/TCSC/UKHEC/GridWG>
- [4] F. Costen, J.M. Brooke, R.J. Allan and M. Westhead *Grid-based High Performance Computing* (UKHEC, 2000) <http://www.ukhec.ac.uk>
- [5] R.J. Allan, S. Pickles, J.M. Brooke, F. Costen and M. Westhead *UK HEC Grid: background and first steps* UKHEC Report to the Research Councils (UKHEC 2000) <http://www.ukhec.ac.uk>
- [6] R.J. Allan, S.J. Andrews, M.F. Guest, P.M. Oliver, D. Henty, L. Smith, S. Telford and S. Booth *Designing and Building Beowulf-class Cluster Computers* (UKHEC 2000) <http://www.ukhec.ac.uk>
- [7] R.J. Allan *Developing a Web Portal for the Computational Grid* Draft Technical Report, UKHEC Collaboration and CLRC e-Science Centre (2001) <http://esc.dl.ac.uk/GridWG>
- [8] 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.
- [9] R.J. Allan, D.R.S. Boyd, T. Folkes, C. Greenough, D. Hanlon, R.P. Middleton and R.A. Sansum *Evaluation of Globus and Associated Grid Middle-ware*. Report to the DTI (CLRC e-Science Centre, May 2001) <http://esc.dl.ac.uk/StarterKit>
- [10] R.J. Allan and D.R.S. Boyd *Some Architectural Issues to be considered in building a UK Grid* Report for the UK Grid Engineering Task Force (CLRC 2001)
- [11] R.J. Allan, D.R.S. Boyd, T. Folkes, C. Greenough, D. Hanlon, R.P. Middleton, R.A. Sansum, Elson Mourão, Rob Baxter, David Henty, Mark Parsons, John Brooke, W.T. Hewitt, Mike Daw, Jon MacLaren, Jon Gibson, Graham Riley and Stephen Pickles *Globus and Associated Grid Middleware. Consolidated Evaluation Report from UKHEC Sites*. (UKHEC, December 2001) <http://esc.dl.ac.uk/StarterKit>
- [12] R.J. Allan, S.J. Andrews and M.F. Guest *HPCGrid* the national magazine for HPC applications (Daresbury Laboratory, twice yearly) ISSN 1473-2076 <http://www.cse.clrc.ac.uk/Activity/HPCProfile>
- [13] M. Baker, H. Ong and G. Smith *A Report on Experiences operating the Globus Toolkit through a Firewall* (University of Portsmouth, 2001) <http://www.dl.ac.uk/TCSC/UKHEC/GridWG>
- [14] 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://www.dl.ac.uk/TCSC/UKHEC/GridWG>
M. Baker, H. Ong and G. Smith *A Portal for Liquid Crystal Modelling Applications* (University of Portsmouth, 2002) <http://www.dl.ac.uk/TCSC/UKHEC/GridWG>

- [15] M. Baker, H. Ong and G. Smith *Grid Monitoring Architecture* draft report (University of Portsmouth, 2002)
- [16] R.J. Allan, D.J. Hanlon, K. Kleese van Dam, S. Sufi, L. Sastry, D.R.S. Boyd and B.M. Matthews *An Integrated e-Science Environment for CLRC: Grid Portals, HPC Computation, Data Management and Advanced Visualisation* Project Overview (CLRC, 2002), <http://www.dl.ac.uk/TCSC/UKHEC/GridWG/>
R.J. Allan, K. Kleese van Dam, L. Sastry, and B.M. Matthews *An Integrated e-Science Environment for CLRC: Grid Portals, HPC Computation, Data Management and Advanced Visualisation* Project Definition, Use Cases and Component Specifications (CLRC, 2002),
- [17] O.F. Rana, V. Getov, E. Sharakan, S.J. Newhouse and R.J. Allan *Developing Grid Services with Jini and JXTA* White paper for Jini Working Group presented at GGF4, Toronto (February 2002) <http://esc.dl.ac.uk/WebServices>
- [18] R.J. Allan, D. Chohan, X.D. Wang and M. McKeown *UDDI and WSIL for e-Science* Grid Support Centre (2002) <http://esc.dl.ac.uk/WebServices>
- [19] 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/WebServices>
- [20] Rob Allan and Xiao Dong Wang *A Review of some Web Services Tools*. Grid Technology Group, Daresbury Laboratory (2002) <http://esc.dl.ac.uk/WebServices>
- [21] R.J. Allan and D.R.S. Boyd *The UK e-Science Grid*. Grid Support Centre (2002)
- [22] R.J. Allan et al. *Level 2 Grid – Final Report* (ETF, April 2003)
- [23] Rob Allan, Xiao Dong Wang and Andy Richards *XML Schema for HPC Applications and Resources* Grid Technology Group, Daresbury Laboratory (2002) <http://esc.dl.ac.uk/WebServices>
- [24] R.J. Allan et al. *Middleware, Applications and Tools deployed at Level 2* Proc. AHM'2003 (Nottingham, September 2003)
- [25] 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)
- [26] A.J. Richards, R.J. Allan, G. Drinkwater and K. Kleese *Advanced Grid Computing* Proc. AHM'2003 (Nottingham, September 2003)
- [27] 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)
- [28] Rob Allan, Chris Awre, Mark Baker and Adrian Fish *Portals and Portlets 2003* Proc. NeSC Workshop 14-17/7/2003. Technical Report UKeS-2004-06. http://www.nesc.ac.uk/technical_papers/UKeS-2004-06.pdf (CCLRC and NeSC, 2004)
- [29] Rob Allan and Rob Crouchley *Roadmap for a UK Virtual Research Environment* JCSR(04)14. Report of the JISC JCSR VRE Working Group (JISC 2004) http://www.jisc.ac.uk/uploaded_documents/VREroadmapv4.pdf
- [30] Mark Baker, Hong Ong, Rob Allan and Xiao Dong Wang *Advanced Portal Services* Proc. AHM'2004 (Nottingham, September 2004) <http://www.allhands.org.uk/proceedings/papers/223.pdf>
- [31] Rob Allan, Rob Crouchley, Adrian Fish and Dharmesh Chohan *Portal Services for Awareness and Training* Proc. AHM'2004 (Nottingham, September 2004) <http://www.allhands.org.uk/proceedings/papers/234.pdf>

- [32] 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
- [33] D. Gannon, G. Fox, M. Pierce, B. Plale, G. von Laszewski, C. Severance, J. Hardin, J. Alameda, M. Thomas, J. Boisseau *Grid Portals: A Scientist's Access Point for Grid Services* GGF Community Practice Document (GCE-RG, 19/9/03) <http://forge.gridforum.org/projects/ggf-editor/document/GCE-Portal-working-draft/en/1/GCE-Portal-working-draft.pdf> <http://www.extreme.indiana.edu/~gannon/ggf-portals-draft.pdf>
- [34] G. Fox and D.W. Walker *UK e-Science Gap Analysis* (NeSC, 2003) http://www.nesc.ac.uk/technical_papers/UKeS-2003-01/
- [35] 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>
- [36] M. Baker, H. Ong and G. Smith *A Portal for Liquid Crystal Modelling Applications* (University of Portsmouth, 2002) <http://esc.dl.ac.uk/GridWG>
- [37] Adrian Fish *ReDRESS Project Review: Survey of Portals and Related Software* (University of Lancaster, February 2004)
- [38] 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)
- [39] 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
- [40] J. Novotny, V. Welch *MyProxy* (NCSA) www.grid.ncsa.uiuc.edu/myproxy/
- [41] *Building Collaborative eResearch Environments* Final report compiled for JISC by Andrew Cox (University of Loughborough, March 2004)
- [42] *UK Role in Open Grid Services Architecture* draft report v0.7 from the Grid Architecture Task Force (12/3/02)
- [43] Liz Lyon and Andy Powell *A Distributed National Electronic Resource Technical Architecture: Scoping the Information Environment* (UKOLN, University of Bath). See also <http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/>
- [44] S. Wilson, B. Olivier, S. Jeyes, A. Powell, T. Franklin *A Technical Framework to Support e-Learning* Report to JCLT (2003)
- [45] R. Crouchley, R.J. Allan et al. *A Service-Oriented Architecture for VRE/ VLE building on the Sakai Framework* Report to JCSR, JCSR(04)14a (2004)
- [46] Mark J. Norton *A Comparison between the JISC and Sakai Frameworks* (24/3/04)
- [47] Michael Stevens *The Benefits of a Service-Oriented Architecture* Developer.com <http://www.developer.com/services/article.php/1041191>
- [48] Michael Stevens *Service-Oriented Architecture Introduction* (2 parts) Developer.com <http://www.developer.com/services/article.php/1010451>
- [49] *Web Services and Service-Oriented Architectures* <http://www.service-architecture.com/>

- [50] Sayed Hashimi *Service-Oriented Architecture Explained* (O'Reilly) http://www.ondotnet.com/pub/a/dotnet/2003/08/18/soa_explained.html
- [51] Bill Ruh *Succeeding at Service Oriented Architecture* (ZDNet) <http://www.zdnet.com.au/builder/architect/work/story/0,2000034884,20276810,00.htm>
- [52] *The Arda Project – a Realisation of Distributed Analysis for LHC*. <http://lcg.web.cern.ch/LCG/peb/arda/Default.htm>
- [53] *OGCE: Open Grid Computing Environment* <http://www.ogce.org>
- [54] D. Gannon et al. *Building Grid Portal Applications from a Web-Service Component Architecture* (Xtreme Computing Laboratory, University of Indiana, 1/4/2004) <http://www.extreme.indiana.edu/xportlets/publications/>
- [55] Web Services interest area of UK Grid Engineering Task Force <http://www.grid.ac.uk/ETF/public/WebServices/index.shtml>
- [56] registry interest area of UK Grid Engineering Task Force <http://www.grid.ac.uk/ETF/public/registry/index.shtml>
- [57] Portals and Grid Clients interest area of UK Grid Engineering Task Force <http://www.grid.ac.uk/ETF/public/Portals/index.shtml>
- [58] Savas Parastatidis *GAF: Grid Application Framework* See <http://www.grid.ac.uk/ETF/public/WebServices/index.shtml>
- [59] Mark McKeown *OGSI::Lite* README file with the Perl source from <http://www.sve.man.ac.uk/Research/AtoZ/ILCT>. See <http://www.grid.ac.uk/ETF/public/WebServices/index.shtml>
- [60] P.V. Coveney and J. Chin *Towards tractable toolkits for the Grid: a plea for light-weight, usable middleware*. <http://www.realitygrid.org/lgpaper.html>
- [61] Bill Olivier *Information Paper for JISC VLE Programme* (JISC, June 2004)
- [62] R.J. Allan *GROWL: Grid Resources on Workstation Library* <http://www.grid.ac.uk/GROWL>
- [63] *JSR-168 Portlet Specification* <http://www.jcp.org/en/jsr/detail?id=168>
- [64] *Pluto* <http://portals.apache.org/Pluto>
- [65] *Jetspeed 2* <http://portals.apache.org/jetspeed-2>
- [66] *uPortal* <http://mis105.mis.udel.edu/ja-sig/uportal>
- [67] *WSRP Web Services for Remote Portlets TC site* OASIS http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp
- [68] *WSRP UDDI Technical Note* OASIS <http://www.oasis-open.org/apps/org/workgroup/wsrp/wsrp-pfb/download.php/7671/wsrp-pfb-uddi-tn-draft-08.pdf>
- [69] *WSRP4J* <http://ws.apache.org/wsrp4j>
- [70] *Swing and WSRP*
- [71] S. Wilson, K. Blinco and D. Rehak *Service Oriented Frameworks* DEST (Australit, JISC-CETIS (UK and Industry Canada http://www.jisc.ac.uk/uploaded_documents/AltilabServiceOrientedFrameworks.pdf
- [72] A. Fish *Sakai Architecture* (University of Lancaster, 2004) <http://redress.lancs.ac.uk/sakai-uPortal>

- [73] A. Fish *Sakai Installation Guide* (University of Lancaster, 2004) <http://redress.lancs.ac.uk/sakai-uPortal>
- [74] *JavaServer Faces* Core book
- [75] *GEMMLCA: Grid Execution Management for Legacy Code Architecture* (CPC, University of Westminster) <http://www.cpc.wmin.ac.uk/ogsitestbed/GEMMLCA/>
- [76] *WSRF::Lite* (University of Manchester) <http://www.sve.man.ac.uk/Research/AtoZ/ILCT>
- [77] *Glite* Lightweight middleware for Grid computing (EGEE Project) <http://glite.web.cern.ch/glite/>
- [78] Integrative Biology Workshop (Oxford, 30/9-1/10/04)
- [79] Mark Norton *Porting tools into Sakai*. <http://chefproject.org/access/content/group/1085757908747-8684/Tutorials/writing-sakai-tools.pdf>
- [80] Copper Core Web site <http://coppercore.org>
- [81] JavaServer Faces <http://java.sun.com/j2ee/javaserverfaces/>
- [82] Maven <http://maven.apache.org/>
- [83] PostgreSQL <http://www.postgresql.org/>
- [84] Sakai Installation <http://cvs.sakaiproject.org/release/1.0.rc2/>
- [85] CHEF <http://chefproject.org/portal>
- [86] ANT <http://ant.apache.org/>
- [87] W. Edwards, and J.R. Newman *Multiattribute Evaluation* (Sage, 1982)
- [88] C. White *Is the Portal Dead?* DMReview Online (July 2003) http://www.dmreview.com/article_sub.cfm?articleID=6959
- [89] G. Phifer *The Portal is Dead, Long Live the Portal!* Gartner Research Review COM-19-4418 (28/2/03)

A WP 1 – Comparing Sakai/ CHEF with Alternative Frameworks for a VRE

This work package compared Sakai/ CHEF with Alternative Frameworks for a VRE. By “framework” we really imply a means to deliver the VRE services in an appropriate way to end users and developers. This was an ongoing task and lasted for the duration of the project. Even though Sakai/ CHEF appears to be the leader as a VRE portal delivery framework at the moment, largely because of the tools it already offers and its use as a VLE, this may change. This work package resulted in the Technology Report (Evaluation Report 1) below. See also [37].

Work carried out includes:

Task	Title	Responsibility	Description
1.1	Portals and Portlets 2003	Daresbury	Review projects presented and surveyed in the Portals and Portlets 2003 NeSC workshop and update material
1.2	ReDRESS TR	Lancaster	Review projects identified in the ReDRESS Technology Survey and update material
1.3	Evaluate new work	Both	Ongoing technology watch for related projects, e.g. by liaison with GGF Grid Computing Environments research group, JCLT VLE groups, FPVI EGEE project developers, etc.
1.4	Compile Technology Evaluation Report	Both	Use material to compile report, keep up to date over time of project on Web sites and provide to directors for submission to JCSR and other appropriate groups, e.g. GGF GCE research group

A.1 Portals and Portlets 2003

We reviewed projects presented and surveyed in the Portals and Portlets 2003 NeSC workshop [28].

An updated list of public-domain and proprietary portal offerings is given in Appendix I.

A.2 ReDRESS Technology Evaluation Report

This report [37] compares Sakai/ CHEF with Alternative Frameworks for a Virtual Research Environment.

We have carried out work to install, evaluate and use Jetspeed, uPortal, GridSphere, eXo Portal, CHEF, OGCE, Sakai RC1, Sakai RC2 and Sakai v1.0.

A.3 Evaluate new Work

Ongoing technology watch for related projects, e.g. by liaison with GGF Grid Computing Environments research group, JCLT VLE groups, FPVI EGEE project developers, etc.

We have evaluated and tested by writing simple portlet examples the Java WSRP4J package to export to a Swing client and to export a portlet developed with Sakai to a uPortal client for rendering.

We have investigated the use of WSRP with Perl, but there are currently no tools available. The fact that SOAP::Lite has not been updated recently would also mean that the Perl SOAP support needed to ingest

WSDL from a WSRP service is lacking. A similar situation is still apparent for C producers and consumers meaning that, although WSRP is claimed to provide a language-agnostic portlet interface, it is at present not very useful. Whilst WSRP4J does however provide a bridge for Java clients and services, it is still evolving.

B WP 2 – Ease of Administration (EoA) of Sakai/ CHEF for a VRE.

This work package assessed the Ease of Administration (EoA) of Sakai/ CHEF for a VRE. Administration facilities in the chosen VRE framework are of paramount importance to its maintenance and extension. There are many questions that we have attempted to answer, e.g. *How easy is it to add new users to Sakai/ CHEF?*, *Can we customise the portal “skin” in appropriate ways for existing e-Science projects?*, *Can we easily allow projects to create work sites for subsets of users in the portal and to include the tools they require from the given set?*, *Can we isolate users from specific work sites and create a secure access mechanism using certificates (e.g. with GridSite)?*

This work package resulted in the production of a report (Evaluation Report 2) together with any software adapters developed during the assessment process. These adapters may be required to plug Sakai/ CHEF into Lancaster/ Daresbury authentication mechanisms and will serve as useful templates for other institutions.

Administration facilities in the chosen VRE framework are of paramount importance to its maintenance and extension. There are questions that we have attempted to answer:

Task	Title	Responsibility	Description
2.1	Add users	Lancaster	<i>How easy is it to add new users to Sakai?</i>
2.2	Customisation	Lancaster	<i>Can we customise the portal “skin” in appropriate ways for existing e-Science projects?</i> demonstrators for ReDRESS, e-HTPX, e-Minerals and NGS
2.3	Worksite creation	Lancaster	<i>Can we easily allow projects to create worksites for subsets of users in the portal and to include the tools they require from the given set?</i>
2.4	Security	Both	<i>Can we isolate users from specific worksites and create a secure access mechanism using certificates (e.g. with GridSite)?</i>
2.5	Performance and scalability	Both	Investigate what server is required to host the VRE front end for a large project, e.g. using BladeCenter at Daresbury

We have additionally included notes on the installation procedures for CHEF and for Sakai RC1 and RC2.

CHEF Installation

CHEF is a predecessor of Sakai. Sakai is also sometimes referred to as CHEF2. The CHEF portal framework is based on the following technologies.

- Jetspeed is an open source implementation of an Enterprise Information Portal, using Java and XML. Chef tools are portlets which run in the Jetspeed environment. The data presented via Jetspeed is independent of content type. This means that content from for example XML, RSS or SMTP can be integrated with Jetspeed. A patched version of Jetspeed portal framework 2002_10_08_patched_jakarta-jetspeed.zip was used in CHEF.
- Turbine is a servlet based framework that allows building of secure web applications quickly. Turbine supports the Velocity or JSP to design the presentation layer.

- Velocity is an open source Java-based template engine used to create templates according to the Model-View-Controller (MVC) model. CHEF uses Velocity to provide template services for Turbine in the Jetspeed environment.

The current public release of CHEF is 1.2.10. The software can be downloaded from <http://chefproject.org/project/src/>. The Chef software was installed on a Linux machine, *thames.dl.ac.uk*, running operating system Redhat WS release 3.

The CHEF portal can be seen at <http://thames.dl.ac.uk:8080/chef/portal>. The installation and deployment of CHEF portal was based on using the following software.

- ANT is a Java based build tool, Ref 6. Version 1.6.2
- PostgreSQL database used for content management. PostgreSQL version 7.4.3.
- Tomcat servlet container to host the Sakai portal. Tomcat version 5.0.27.

As with Sakai, CHEF by default only supports the MySQL and Oracle databases and changes were made to the SQL scripts in order to make CHEF work with the PostgreSQL database.

The installation and setting up of the PostgreSQL database is similar in Sakai. Before populating the CHEF database, the following changes need to be made to the SQL scripts,

- Change Mediumtext to Text
- Comment out "DROP TABLE IF EXISTS ..."
- Change Datetime to Timestamp
- Change Integer unsigned (0 to 4294967295) to Serial

Sakai RC1 Installation

Our documented installation procedure for Sakai is also available as a separate report [73]. It gives step by step instructions regarding the installation of a Sakai instance at the reader's institution. The document is aimed at a technical user with some experience in Apache Tomcat and PostgreSQL RDBMS administration.

Java Installation

1. Go to the Sun Java website (<http://java.sun.com/j2se/1.4.2/download.html>) and download the Java SDK (Software Development Kit).
2. Make the downloaded file executable and run it in the directory that you want the SDK installed in (I use '/usr/local').
3. Create an environment variable called JAVA_HOME pointing to the directory that you just created in step 2. I export the environment variable in '/etc/profile'.

Maven Installation

1. Go to the Apache Maven site (<http://maven.apache.org/start/download.html>) and download Maven.
2. Expand the Maven archive into a directory (I use /usr/local). Create a symbolic link called maven pointing to the newly created Maven directory.

3. Add `’/usr/local/maven/bin’` to your PATH environment variable.

Install and Configure Tomcat

1. Go to the Apache Jakarta website (<http://jakarta.apache.org/>) and download Tomcat (I used versions 5.0.25 and 5.0.28 for the evaluation ,they both work OK).
2. Expand the tomcat archive into a directory (I use `’/usr/local’`)
3. Create a symbolic link to the expanded directory called `’tomcat’`, the expanded directory will be called something like `’jakarta-tomcat-5.0.28’`.
4. Create an environment variable called `’CATALINA_HOME’` pointing to the symbolic link that you just created in step 3. My entry is `’CATALINA_HOME=/usr/local/tomcat’`.
5. Create an environment variable called `’CATALINA_OPTS’` and set it to `’-Xms128M -Xmx256M’`. My entry would be `’CATALINA_OPTS=-Xms256M -Xmx512M’`. The double quotes are essential. This permits the Java runtime to use up to 256MB of heap memory. Tomcat needs a LOT of heap memory, so if you have more memory available, increase the `-Xmx` part to half of what you have. A good rule of thumb is to use two thirds of what is available on a server that is mainly using tomcat, and half if it is a workstation being used for other purposes.

Download Sakai

1. Go to the Sakai website (<http://cvs.sakaiproject.org/release/1.0.b1/src/sakai-src.zip>) and download the sakai source archive to your computer.
2. Expand the archive into `’HOME_DIR/projects/sakai’` or something similar, this directory will be referred to as `’SAKAI_SRC_DIR’` for the duration of this document. I use `’/home/fisha/projects/sakai’`.

Download, Install and Configure PostgreSQL

Download

1. Download the PostgreSQL source from <http://www.postgresql.org>
2. Expand the source into your home directory. For me this would be `’/home/fisha/src’`.

Install

1. Follow the instructions in the expanded source. They are in a file called `INSTALL`. When you have completed this process you will hopefully have a working PostgreSQL database server on your computer.
2. `su` to postgres (`’su -l postgres’`) and create a `’sakai’` user. You do this by running the command `’createuser -P sakai’`. You will be prompted for a password. I will refer to the sakai database username and password from hereon as `’SAKAI_DB_USER’` and `’SAKAI_DB_PASSWORD’` respectively. When asked if the sakai user needs to create databases and users, say no.
3. Create the Sakai database. As the postgres user, run the command `’createdb -O SAKAI_DB_USER sakai’`. This means `’create an empty database called sakai, and mark it as being owned by the user ’SAKAI_DB_USER’`.

Configure Sakai Database

Sakai comes with a set of SQL scripts for building the Sakai database. Unfortunately, these are tuned for Oracle and thus need modifying before running across PostgreSQL.

1. Change directory to 'SAKAI_SRC_DIR/deploy/src/sql/legacy'.
2. Copy the entire oracle directory to a new directory called postgres, then change to that directory..
3. Concatenate all the files beginning with 'chef_' into one file called 'pg_all.sql'.
4. Edit pg_all.sql and delete all the lines starting with 'DROP'.
5. Replace all instances of 'VARCHAR2' with 'VARCHAR'.
6. Replace 'BODY LONG RAW' with 'BODY BYTEA'.
7. Replace 'SESSION_USER' with '"SESSION_USER"'. This is a reserved word in PostgreSQL so it has to be wrapped in double quotes for the database server to accept it.
8. Switch user to postgres ('su -l postgres') and create the sakai tables by running 'psql -U SAKAI_DB_USER SAKAI < pg_all.sql'.

Build and Deploy Sakai

1. Copy the directory 'SAKAI_SRC_DIR/deploy/src/usr_local_sakai' to '/usr/local/sakai'.
2. Change directory to '/usr/local/sakai'.
3. For each directory 'sakai-component', 'legacy-component', and 'framework-component', change to the directory, delete 'components.xml' and then rename 'components_db.xml' to 'components.xml'.
4. Change directory back to '/usr/local/sakai' and edit 'sakai.properties'.
5. Locate the lines specifying the Oracle database parameters and cut and paste them back into the document. Change the 'sql.driver' entry so that it reads 'sql.driver=org.postgresql.Driver'. Change the 'sql.connect' entry so that it reads 'sql.connect=jdbc:postgresql://localhost:5432/SAKAI'. Change the 'sql.user' entry so that it reads 'sql.user=SAKAI_DB_USER' (remember, SAKAI_DB_USER refers to the PostgreSQL user that owns the sakai database). Change the 'sql.pw' entry, so that it reads 'sql.pw=SAKAI_DB_PASSWORD'.
6. Change directory to 'legacy-component' and edit 'components.xml'. Wherever there is a property named 'locksInDb', change the value from true to false.
7. In the directory 'SAKAI_SRC_DIR/legacy-component/src/java/org/sakaiproject/component/legacy/id' edit 'ClusterIdService.java', replacing 'CHEF_ID_SEQ.NEXTVAL from dual' with 'nextval('CHEF_ID_SEQ)'.
8. In the directory 'SAKAI_SRC_DIR/legacy-component/src/java/org/sakaiproject/component/legacy/event' edit 'ClusterEventTracking.java', replacing 'CHEF_EVENT_SEQ.NEXTVAL from dual' with 'nextval('CHEF_EVENT_SEQ)'.
9. In the directory 'SAKAI_SRC_DIR/framework-component/src/java/org/sakaiproject/component/framework/session' edit 'ClusterUsageSessionService.java', replacing 'SESSION_USER' with '"SESSION_USER"'. SESSION_USER is a reserved word in PostgreSQL (unlike Oracle) and thus needs to be enclosed in escaped double quotes.

10. Setup the 'build.properties' file up in your home directory, as per the instructions at the sakai download site. The 'maven.repo.remote' entry can be modified to add another repository and thus speed project building. I added 'http://mirrors.sunsite.dk/maven' to the comma separated list of repositories.
11. Change to SAKAI_SRC_DIR and type 'maven'. Sakai should compile and a set of web applications will be installed into Tomcat. You should now be able to go to the URL 'http://localhost:8080/sakai-uPortal' and log in as 'admin' with password 'admin'. You obviously need to change this as soon as possible!

Differences with Sakai RC2 Installation

The current public release is Sakai 1.0 RC2 (Release Candidate 2). The software can be downloaded from <http://cvs.sakaiproject.org/release/1.0.rc2/>. The Sakai software was installed on a Linux machine, trent.dl.ac.uk, running operating system Redhat WS release 3. The Sakai RC2 portal can be seen at <http://trent.dl.ac.uk:8080/portal>. The build and deployment of Sakai portal was based using the following software.

- Maven is a project build tool, Ref 2. Release 1.0-RC4.
- PostgreSQL database used for content management. PostgreSQL version 7.4.3.
- Tomcat servlet container to host the Sakai portal. Tomcat version 5.0.27.

The current Sakai release only supports the MySQL and Oracle databases. Some changes were made to the Sakai source in order to make it work with the PostgreSQL database.

The instructions for setting up the build environment and configuration files can be found in [84].

Sakai looks in files on the server to get configured in `/usr/local/sakai`.

PostgreSQL Setup

Install and setup a database for Sakai use according to PostgreSQL instructions, making sure the file `pg_hba.conf` is correctly setup for database access.

Database changes:

Edit the file which stores the session information in the database, `/usr/local/sakai-dev/deploy/src/sql/legacy/oracle/chef_session.sql`. The variable `SESSION_USER` is reserved in PostgreSQL. The word `SESSION_USER` should be placed in quotes, ie `"SESSION_USER"`.

Change the following Oracle specific syntax to PostgreSQL syntax in `/usr/local/sakai-dev/deploy/src/sql/legacy/oracle/chef_*.sql`,

- long to text
- Varchar2 to Varchar
- Comment out `"DROP TABLE IF EXISTS ..."`

Populate the Sakai database with default information by running the file `/usr/local/sakai-dev/deploy/src/sql/legacy/oracle/all.sql` script.

Modify file in `/usr/local/sakai/legacy-component/components.xml` and change `locksInDb` to false.

To work with PostgreSQL database, the file in `/usr/local/sakai.properties` needs to be modified to add the database driver settings as follows,

```
sql.driver=org.postgresql.Driver
sql.connect=jdbc:postgresql://localhost/<sakaiDB>
sql.user=<sakai>
sql.pw=<sakai>
```

Code changes:

Change all reference to `SESSION_USER` to `SESSION_USER` in java file `ClusterUsageSessionService.java`.

Some of the SQL syntax is specific to Oracle and therefore requires changes in the Java source as follows. Note the location of source files is in `$SAKAI-DEV/` where `$SAKAI-DEV` is the directory path of Sakai installation.

- Change `CHEF_EVENT_SEQ.NEXTVAL` to `nextval('CHEF_EVENT_SEQ')` in `ClusterEventTracking.java`
- Change `"CHEF_ID_SEQ.NEXTVAL from dual"` to `nextval('CHEF_ID_SEQ')` in `ClusterIdService.java`
- Remove the word `"nowait"` in `BaseDbSingleStorage.java`

The PostgreSQL JDBC jar file needs to be copied to `$CATALINA_HOME/common/lib`.

Installation using CVS Version

Note Sakai RC2 was released with JavaServer Faces integration somewhat broken. A much improved source was obtained from the current CVS to do the Technology Portability Profile (TPP) tool development using JSF. The CVS version consists of on going development work to the Sakai framework and hence the software should be considered with some risks, as the enhancements or new features will not have been tested fully.

The above changes to PostgreSQL apply when using the CVS version to set up the Sakai portal.

B.1 Add Users

In addition to using the built-in CHEF and Sakai configurations in which it is very easy to add new users, we carried out two tests to show that CHEF and Sakai can be integrated with existing user management systems.

Lancaster AuthN Integration

CHEF's login mechanism has been modified to use Lancaster University's LDAP directory interface for the primary means of authentication with the CHEF internal database-backed system being used as a secondary method. This means that any user, be they student or staff, can log straight into CHEF using their standard network username and password. LDAP is a very commonly used to hold institutional people data, such as names, addresses, roles and passwords and many tools exist to handle LDAP data, often by the LDIF conversion format. Because of this familiarity the Globus Grid middleware developers used it for their MDS information model with referral between multiple LDAP directories.

The piggy backing of an LDAP directory lookup on CHEF's default mechanism like this means that we can also authenticate external ReDRess users by adding them to the CHEF user database. The CHEF documentation gives some guidance on how to achieve this. This is what we did:

1. Created a java class that implements the interface *PiggybackUdp*. You need to implement all of the methods, including the lookup methods, so that they query your institution's LDAP directory implementation. The *Udp* doesn't just authenticate users, it is used during worksite creation, see Section B.3, to check that the participants are known to the CHEF software. This gives a degree of referential integrity to the worksite creation process, so this is why the lookup methods are also important. The CHEF documentation recommends that this go in a 'plugin' sub-package.
2. Modified the 'compile' target in CHEF's `build.xml` file so that the new class is compiled into the CHEF Web application's class tree.
3. Modified the `chef_dev_nc_resources.properties` file in `src/conf` by changing the plugin component class *SampleUserDirectoryProvider* to the fully qualified name of the newly implemented class.

Lancaster Student Records System Integration

Lancaster uses a student records system called LUSI. This is based on a Microsoft SQL database with a Windows front end for report presentation. It was not possible to gain direct access to this system for the evaluation, so a similar test database has been created. This was linked to a CHEF service called *StudentRecordsService* which has three methods, *getCourseList()*, *getStudentList()* and *getTeacherList()*, the latter two of which take a course id as their sole parameter. This was wired into the Turbine framework using the CHEF configuration files. To permit the adding of a course worksite the *UmiacClient* functionality was replaced with calls to the custom *StudentRecordsService*. This was done, within the *SiteAction* class, by modifying the *finishCourseSite()* and *addNewSite()* methods.

The ability to link into people records systems such as this would be very beneficial in the creation of a VRE. Many researchers access large-scale facilities, such as ISIS, SRS, CLF and HPCx at CCLRC or Jodrell Bank and CSAR at Manchester. Currently all these facilities have diverse records systems for their users.

B.2 Customisation

CHEF was re-skinned to more closely resemble a Lancaster University Web site and initially for the ReDRESS project. A version has also been re-skinned for the NGS. ReDRESS is now using Sakai RC1. This re-skinning has been lightweight in that none of the layouts have been changed, only colours and graphics such as the logo used on the page template.

B.3 Worksite creation

CHEF and Sakai have been presented and demonstrated at meetings with members of the following projects: ReDRESS, NCeSS, e-HTPX, e-Minerals, MyGrid, RealityGrid, NGS and Integrative Biology. During these demonstrations the worksite setup and user management procedures have been demonstrated on-line showing that it can be easily achieved. The ability to organise the portal into project-based worksites and select the users accessing each site and the tools available to them is deemed to be one of the advantages of the CHEF or Sakai frameworks and effectively differentiate them from others such as GridSite.

B.4 Security

Can we isolate users from specific workspaces and create a secure access mechanism using certificates (e.g. with GridSite)? This has not been tested.

B.5 Performance and scalability

Investigate what server is required to host the VRE front end for a large project, e.g. using the BladeCenter at Daresbury.

In practice performance has not been seen to be a problem with thousands of users of institutional portals based on CHEF.

C WP 3 – Making existing VRE (Grid) Components available in Sakai/ CHEF.

This work package was aimed at establishing the feasibility of making existing VRE (Grid) components available in Sakai/ CHEF. The project partners at Lancaster and Daresbury had access to a range of research tools that could be integrated into a VRE framework. We investigated the feasibility of creating an integrated suite of such tools, in order to determine whether there were any generic problems we needed to forewarned about. This work package contributed to a report (Evaluation Report 2) together with any software wrappers employed in making Grid tools available in the Sakai/ CHEF framework. These wrappers will serve as useful templates for other Grid tools.

Work was be done on the construction of a demonstrator, with a number of such tools integrated. It was anticipated that the construction of such a demonstrator will be the optimal way to establish feasibility and get feedback from users in sample e-Science projects. This work package is closely coupled with WP 4, but we list the components here.

Work carried out includes:

Task	Title	Responsibility	Description
3.1	InfoPortal	Daresbury	Grid Information Services and XML metadata about projects, users, applications and resources
3.2	HPCPortal	Daresbury	“Active” services for authentication, file management, data management, workspace, job submission and applications
3.3	CCF White-board	Daresbury	Collaborative Computing Framework being deployed in the e-Minerals project from U. Reading (UK) and U. Emory (USA). Collaboration with Vassil Alexandrov (Reading)
3.4	SPP Cross search	Lancaster	Subject Portals Project portlet for RND resource cross searching. Collaboration with SPP project developers (Bristol, Bath, Oxford)

Sakai is an ambitious project which already comprises a rich set of tools and templates for extending the set. Since the Denver conference, Mark Norton has taked the talk section on cloning tools and written it up into a full tutorial document [79]. This tutorial contains step-by-step instructions on cloning an existing Sakai beta tool (sakai-module) and modifying it to be a new, stand alone tool. Whilst the example is a very simple “Hello World” application, it can serve as the basis for creating new Sakai tools. Chuck Severance demonstrated this by creating a MyProxy tool.

Pre-built collaboration and research tools have been described in Section 5. Sakai also has a comprehensive and growing set of VLE tools developed in the Sakai Educational Partners’ Programme, see <http://www.sakaiproject.org>.

Discussions have indicated a number of additional tools which could (should) be developed and integrated into a Sakai-based VRE framework. We are aware that work is ongoing both in the UK, USA and elsewhere to develop appropriate open-source Java-based tools in many of these areas which could be adapted quickly.

Further information about this is provided in Appendix F.

Search Tools: Context-based Information Retrieval, Collection Cross Search, Google Web service interface, Application Discovery, Scientific Data Query, UDDI, other information management tools as appropriate;

Collaboration/ Discussion: IRC (Internet Relay Chat), Blog, Address Book, Wiki, Research Scheduler, Video Conferencing, Visualisation, Distributed Whiteboard, Flowchart Editor, Distributed Display, Graphing/ Charting;

Analysis Tools: Scientific Calculator, Active Spreadsheet, Statistical Computing, Data Management, Simple Visualisation, Grid Information, Network Information, Grid computing;

Publication/ Documentation: Text Editor, Word Processor, Distributed Presentation Tool, Bibliography Builder, XHTML Editor, LaTeX Processor, Simple Interactive Plotting, Data and Metadata Upload and Harvesting, Project Publication, Application Publishing, UDDI;

Others: Portal Statistics, Personal Information Manager (Chandler?), e-Notebook, Shopping Cart, Bugzilla, CVS, Portal Workspace, Authorisation Policy Management;

Training and Awareness: Personal Information Guide (PIG), CopperCore Content Player, Tool Templates, Documentation Manager, Helpdesk interface, other e-Learning tools as appropriate.

We report on progress with evaluation and development of a couple of the tools for Sakai. The outcome of these initial evaluations is promising as we are gaining increasing knowledge of working with the CHEF framework code base. The release of Sakai 1.0.b2 on 23rd July and later RC2 enabled us to continue with Sakai and provide a fuller report on this work at the e-Science All Hands Conference in September.

C.1 InfoPortal

InfoPortal has Grid Information Services and XML metadata about projects, users, applications and resources. InfoPortal and HPCPortal outcomes are discussed together below as this has become a separate project to implement a single portal for NGS users.

C.2 HPCPortal and NGS Portal

A National Grid Service portal is now being developed using the CHEF framework for use in a production environment. The NGS portal will consist of the standard CHEF collaboration tools such as Welcome page; Announcements for posting current, time-critical information; Schedule for posting and viewing deadlines, meetings, events, etc; Resources for posting documents, URLs, etc; Discussion for conversation in written form; and Work Site to allow NGS members to setup their own project worksite either as a private Web page or allowing other members to participate in their workspace forming a VO. In addition, the Grid tools are being provided to allow users to perform activities such as Grid Proxy Delegation for creating the x.509 proxy certificate; Grid Resource Broker for job submission; LDAP Browser for resource discovery and GridFTP for file/ data transfer between the NGS compute and data nodes. Examples of these Grid tools had been developed as xportlets by the OGCE team [53, 54].

HPCPortal has “Active” services for authentication, file management, data management, workspace, job submission and applications.

One aim of the ReDReSS project is to access Grid resources to enable live demonstrations of the sort of applications social scientists might want. We are also developing a prototype portal for the National Grid

Service (NGS) to provide easy access to resources at CCLRC and the Universities of Leeds, Oxford and Manchester. To enable this we have ported a couple of tools from InfoPortal and HPCPortal [26] and OGCE into the CHEF framework. This was made easier by the fact that the US NMI portal, OGCE [53], is already using CHEF.

URL feeds from the InfoPortal map and resource status display panels were easily added to CHEF and retained all original functionality. Active services previously found in HPCPortal were included by porting the relevant portlets from OGCE. A portlet was also written from scratch as a client interface to web services in InfoPortal. These return a list of available compute resources on the NGS and also provide query interfaces to return an XML description of a resource (static data) or output from the Globus MDS giving the status of a resource (dynamic data). In this way we have shown that a new Grid tool can be easily integrated into the CHEF/ Sakai portal by adding the tool definition into the XML site description table, adding the tool portlet in the CHEF/ Sakai portlet registry and finally deploying the tool portlet on Apache Tomcat.

Building Grid Tools into Sakai

The CVS version of the Sakai portal framework was used in creating and implementing a simple grid Sakai tool to retrieve an X509 certificate proxy from a Myproxy server.

Building of a Sakai tool depends on understanding how Maven works [82] to manage its source, compile and deploy the application. A knowledge of JSF is required or some understanding of how the Model-View-Controller (MVC) pattern works. This will help the developer in designing the user interfaces and the necessary managed beans (Java methods) which interact with the GUI.

The current best practice to create a Sakai tool is to "clone" an existing tool module, see [79]. This way the directory structure and resource files are kept intact and the developer can use the skeleton structure to develop the new tool. The following outlines some of the steps which would be carried out to convert the clone tool to a new Sakai tool.

- Firstly, copy Sakai module in \$SAKAI-DEV to a new tool <sakai-tool-name>.

An overview of the basic directory structure of Sakai module is as follows:

```

module
  |__ maven.xml
    |__ project.xml
    |__ target
    |__ src
      |__ reg (contains the XML tool registration description)
      |__ bundle (message propertie file for JSF view and
                 internationalisation)
      |__ webapp (web deployment files)
          |__ WEB-INF (web.xml, faces-config.xml)
          |__ JSP files (JSF user interface files)
      |__ java
          |__ component (interface class implementation)
          |__ service (interface service class)
          |__ tool (tool logic application)

```

- Delete the unused files from above directory structure.
- To change the identity of sakai module. – Edit project.xml

- Change the <name>
- Change the <id> – Edit the web.xml file
- Change the <display-name>
- Change the description
- Edit project.xml – Rename file in /reg/sakai.module to sakai.<name>
- Set id to sakai.<name>
- Set url to /sakai-<name>/mytool/jsf.tool
- Edit the JSF file(s) in the webapp directory to define the user interface.
- Edit the file index.html in the webapp directory to change the title and header and replace all sakai-module with sakai.<name>.
- Replace the unwanted tool application logic with the required tool code.
- Rename tool file name with the new tool name.
- Edit the deployment project.xml to add the new tool dependency.

As it can be seen the process of creating a tool requires a number of steps to be completed in describing the build and deployment configuration. There is also an additional effort in implementing the tool logic. However it might take a long time to produce a tool but in the longer run it will be much easier to manage the tool application as the whole process of creating the tool separates out the developing tool logic, user interfaces and configuration.

Building of a Sakai tool is well documented in a technical report "Building a Sakai Tool" by Chuck Severance and Glenn Golden of Sakai Project [79].

C.3 CCF Whiteboard

The Collaborative Computing Framework was being deployed in the e-Minerals project from U. Reading (UK) and U. Emory (USA). This is a collaboration with Prof. Vassil Alexandrov (Reading).

Unfortunately, the port of CCF Whiteboard software into a portlet could not be completed as development of the software has not been continued and the e-Minerals project is no longer associated with it. Instead a portlet tool has been developed to interface to CopperCore (see below).

C.4 SPP Cross search

Subject Portals Project portlet for RND resource cross searching. Collaboration with SPP project developers (Bristol, Bath, Oxford).

Discussions with Jasper Tredgold in Bristol have indicated that the SPP Cross Search portlet will soon be available as open source, but it could not be used during the evaluation work.

C.5 CopperCore Integration

In order to substitute for the evaluative porting of CCF and SPP tool, CopperCore was used instead. CopperCore is available as open source.

CHEF has been loosely coupled with the CopperCore IMS Learning Design player from the Open University of the Netherlands. CopperCore is implemented as a set of enterprise Java beans, and as such requires a bean container to run. JBoss is the bean container of choice for the CopperCore team and this is what we installed. We followed the instructions on the CopperCore Web site [80]. Our aim was to enable the attachment of a number of units of learning to a CHEF course worksite for ReDReSS, via a Web-based tool. The mechanisms employed to do this involved JNDI lookups on the JBoss component registry, to retrieve the appropriate CopperCore stub objects; coupled with extensions to the CHEF worksite modification tool *SiteAction.java* that used these stubs to access the CopperCore engine. We modified the CHEF worksite tool by adding a link pointing to a template containing the upload form. We then added a command to the *SiteAction* class to take the uploaded file, save it to disk, and then pass the file to the CopperCore *LDCourseManager* stub for publishing. To enable the transparent addition of all the “access” level CHEF worksite participants to the ‘learner’ role on each unit of learning we had to modify the *LDCourseManagerBean* and its corresponding interface *LDCourseManager* to enable the retrieval of the learner role ids for a supplied Uol (Unit of learning) id. The final stage of the exercise involved the implementation of a worksite tool enumerating the available units of learning and allowing the subsequent playback of the learning material.

This tool have now been demonstrated at a number of workshops, such as at the NCeSS Training and Awareness Day, Manchester, 19th November 204.

D WP 4 – Issues involved in extending the Functionality of Sakai/ CHEF

This work package was aimed at establishing the issues involved in extending the functionality of Sakai/ CHEF particularly to use Web services for distributed development and deployment. There are some key VRE questions that need to be clearly formulated and which, to our knowledge, are outside the scope of the current Sakai project. One involves the skill set required to extend Sakai/ CHEF functionality and whether this is available in the UK. The degree to which Sakai/ CHEF is standards compliant also needs to be investigated, with a particular attention to standards relating to integration of external tools and content. Sakai is focussing on the OKI OSIDs, but will need to also work with other interfaces. It is anticipated that there will be an inverse relationship between such compliance and the need for substantial extensions to the framework software. A final and key question is how Sakai can be used as a front end to distributed services as envisaged in a service-oriented VRE. This work package contributed to a report (Evaluation Report 2) and developed an architecture as explained in Section 6 and prototype software wrappers to extend the functionality of the Sakai/ CHEF framework. These wrappers could serve as useful templates for other extensions.

The architecture has been proposed following discussions with Sakai technical director Charles Severance and others and was partly instantiated for some of the test services listed in WP 3, e.g. the InfoPortal Web service portlet. The outcomes of these discussions are described along with our conclusions in Section 6.

Work carried out includes:

Task	Title	Responsibility	Description
4.1	WS Architecture	Both	Develop and agree WS-based Open Service Architecture linked to Sakai. Collaboration with Charles Severance (U. Michigan, USA). Input from Geoffrey Fox (U. Indiana) and other members of the GGF Grid Computing Environments research group.
4.2	Implement	Both	Instantiate this architecture to access some of the Grid tools in WP 3.
4.3	Technical Spec	Both	Write up technical specification of this work package so that other services can be incorporated via portlets

An initial technical specification following a number of discussions with relevant developers was presented to the JCSR Committee in a paper on 16th June 2004 [45]. The outcomes of additional discussions with the Sakai developers and the Indiana Xportlets group are described below.

Based on this and work on the ReDReSS and other e-Science projects has indicated the need for extensions to the existing Sakai framework and tools additional to those already provided in CHEF/ Sakai. Our bids to the JISC VRE call focused on achieving these goals and importing a large suite of tools to create a VRE.

This report has focussed on CHEF/ Sakai as a user-facing delivery mechanism for services in a Virtual Research Environment. It has described the underlying Java technology and how it is evolving. The portal framework is however only one component of the e-Research environment. During the course of our work we have therefore needed to address other aspects of VRE architectures which dictate how Sakai or any other 2nd generation portal framework is to be used alongside other components and delivery mechanisms in a Grid such that the many and diverse services supported by JISC and research institutions can be made available.

VRE Architecture

We provide a top-down approach to a possible VRE architecture.

Figure 8 shows how major components in a federated VRE architecture could be linked. In developing this architecture we coined the acronym HIVE: *Highly Integrated Virtual Environment*. This inherits many aspects from CCLRC's prototype Integrated e-Science Environment (IeSE) [16, 26].

Notes:

User/ Application: Consumer of delivered services via tools and applications;

Tool Server: User facing part of the system. Browser, programmig library, desktop icons etc.

Tool Host: The tools server can be Web or desktop based. It will delegate authentication to HIVE server and thus permit single sign-on across remote toolsets;

HIVE Server: HIVE server provides access to integration services such as authentication, workflow, registries. It can handle federated services;

Shibboleth Server: Will provide the authentication services to the HIVE server. It could be part of a federation and thus provide trust-based access to all the tools hosted for all researchers in the federation's institutions;

VO Management: Provides information about users, their roles and project affiliation. It can extend to resources and services;

Registry: Registry holds details of services and provides template to access them along with relevant semantic information. There may be a number of registries handling different types of services. ETF's UDDI and JISC's IESR are examples;

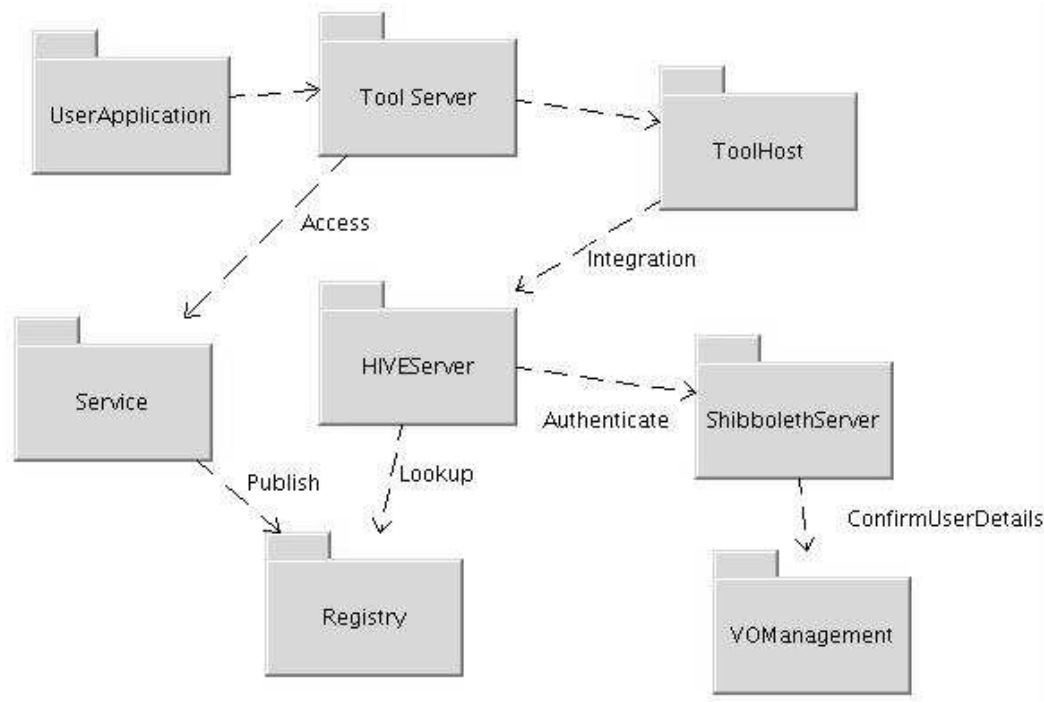


Figure 8: Federated Components in a VRE

Services: Multiple services provide access to end resources and applications. They should be language agnostic and can wrap heritage applications and facilities.

In this architecture there can be multiple instances of each component serving slightly different functions. Web or Grid services using simple protocols are used to connect components with more complex protocols for data delivery.

Integration Services

Within a portal or other framework a number of internal services are needed to address issues of coordination of tools (portlets) within an overall framework. Methods can be provided as an “internal” class library which sits alongside the portlet API and service APIs (the model part of the MVC paradigm). Each framework could have the same or a different set of tools, but the way they are integrated may differ between user groups – similar services are required to allow different frameworks to inter-operate. These services could be federated and available via Web services calls to specialised servers elsewhere in a Virtual Research Environment.

Research issues are implied with most of these services. Some simple ones, such as managing the look and feel of the portal, personalisation and accessibility are provided directly by the portlet container. This is a specialisation of portals and not required in other frameworks. Services which are not specialised, e.g. single sign-on, should not be limited to the portlet API. Some example integration services are now listed:

Session Management: management of a Session Key and issues related to single sign-on and session activities. Involves database access for storing and retrieving other items relevant to the session. User can authenticate and start a new session or revert to a previous one. Service can open and close sessions and log state of a session from state handle. Rollback and replay including personal workflow can be available.

Authentication using MyProxy: MyProxy is a repository of valid proxy certificates for authenticated users. The portal can download these for delegation to trusted external services. Service can also check that certificates etc. are still valid and refresh them if not. Part of the integration API would enable storing and retrieval of the proxy in the portal database for later use. This will be done using the Session Key and uid (e.g. DN or unique e-mail address). Having the cert associated with the session key enables authorisation issues to be tackled, e.g. using subsidiary cert or other method. Same user but different session => working in different role?

VO Management: a virtual organisation could be based around a project (as described by UDDI [18]) which would typically have its own portal and mini-Grid. VO users are real people who have been authenticated and have received a digital identity (certificate). They are then given rights based on the roles they are taking in this VO and thus can be authorised to access services. A prototype schema is given in a separate paper [23].

Integrated State: manage database storage and retrieval of state information by portlet id. Research needs to develop concepts of integrated state. State can be used as an event trigger. State needs to be logged for session manager/ workflow. What states can portlets and services have which are meaningful for rollback and replay?

Service and Portlet Location: registry input, query and lookup of remote services and portlets. This requires semantic support, i.e. what does the service do and why?. It also supports identification and location of internal portlets with unique keys (portlet id) for use in IPC etc.

Portal Preferences: build up a "preferred set" of services, portlets etc. based on usage, e.g. from registries. This service can also log semantic information and build a related ontology. It extends the idea of a workspace toolset enabling dynamic and semantic/ function-driven choice.

Semantic/ Ontology Support: semantic and knowledge-based information about services and portlets in the framework. These are used for decision support and choice augmenting stored preferences. This does not cover generic semantic issues which would need separate tools.

Workflow: directed links between components (typically graph based). Event mechanism used to trigger actions within portals and attached services. Graphs in the portal will be mostly pre-defined, but with constrained facilities to swap in and out components and provide additional inputs at decision points. Again, not completely generic workflow, but based on instances in the preferences list and their states.

Trails and Personalisation: logging of usage for off-line mining and analysis, e.g. for developers to improve presentation, ease of use, and optimisation, e.g. by aggregation of low-level tools and services. This tracks state and component changes for each session. Basically same information as session management service.

IPC - Inter Portlet Communication and Event Management: a message-based communication mechanism between portlets, possibly with event triggers and asynchronous handler. Could be interrupt driven? Use a message queue in the database with associated triggers. Example: Completion of a computational job or background query triggers portlet to send SMS or SMTP message or view results. Could therefore give the user a flag, a chat entry, mobile phone message or an e-mail.

Some key research issues in implementing the above services include:

Identification of user/ session/ portlet/ services: name value pairs => sessionKey, uid, portletId, serviceId. Same stuff as typically put in a cookie in 1st generation portals. Can however also be used by non-portal based tools by building into the method calls.

State definition: a pre-defined set of states needs to be identified. Can this be done? This could be the key to using the event mechanisms and session logging. Do portlets and/ or services make clear state changes?

Extending the Portal Architecture

This evaluation and consideration of the wider implications and how to implement a Virtual Research Environment for the UK have raised some interesting architectural questions. Similar questions have been raised within the JISC VLE programme, e.g. see the paper by Bill Olivier [61]. The key to both VLE and VRE deployment is to ensure the maximal use of existing resources via a re-usable set of distributed services delivered through a variety of mechanisms such as portals and desktop applications.

Figure 9 shows a modification of the portal tool pluggability picture from Charles Severance. It gives the additional API for the integration services and link to remote resources via Web services.

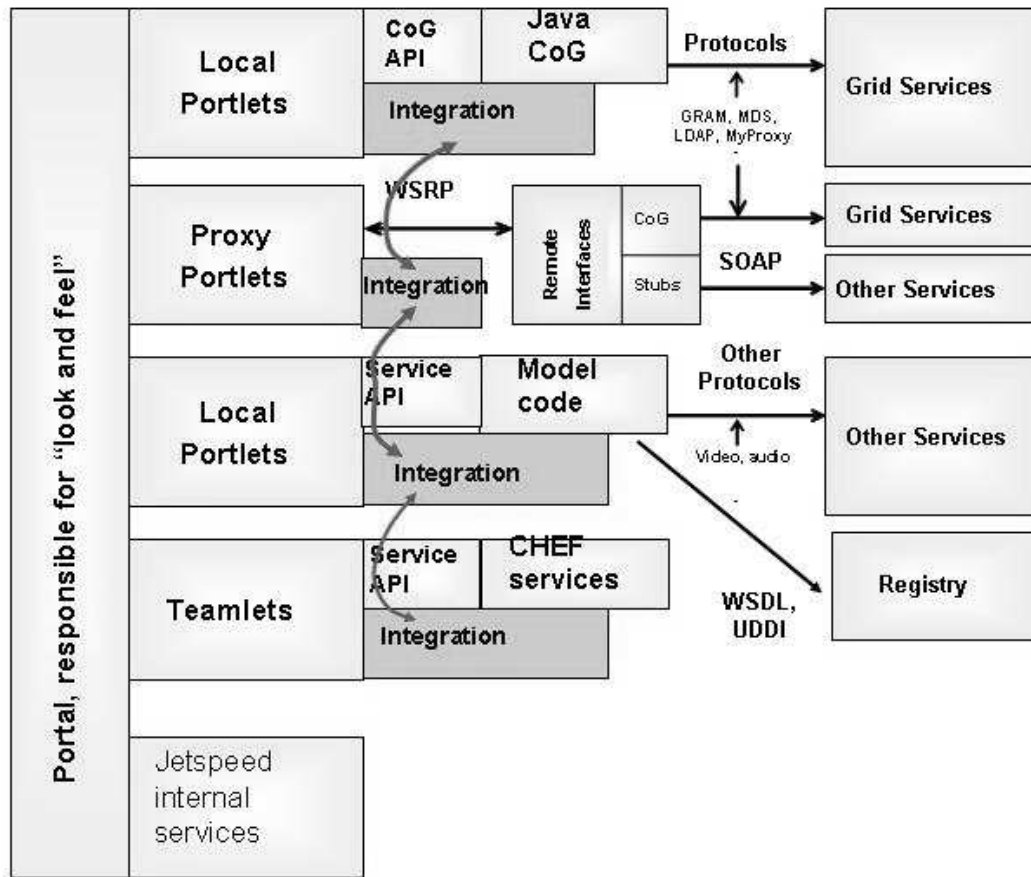


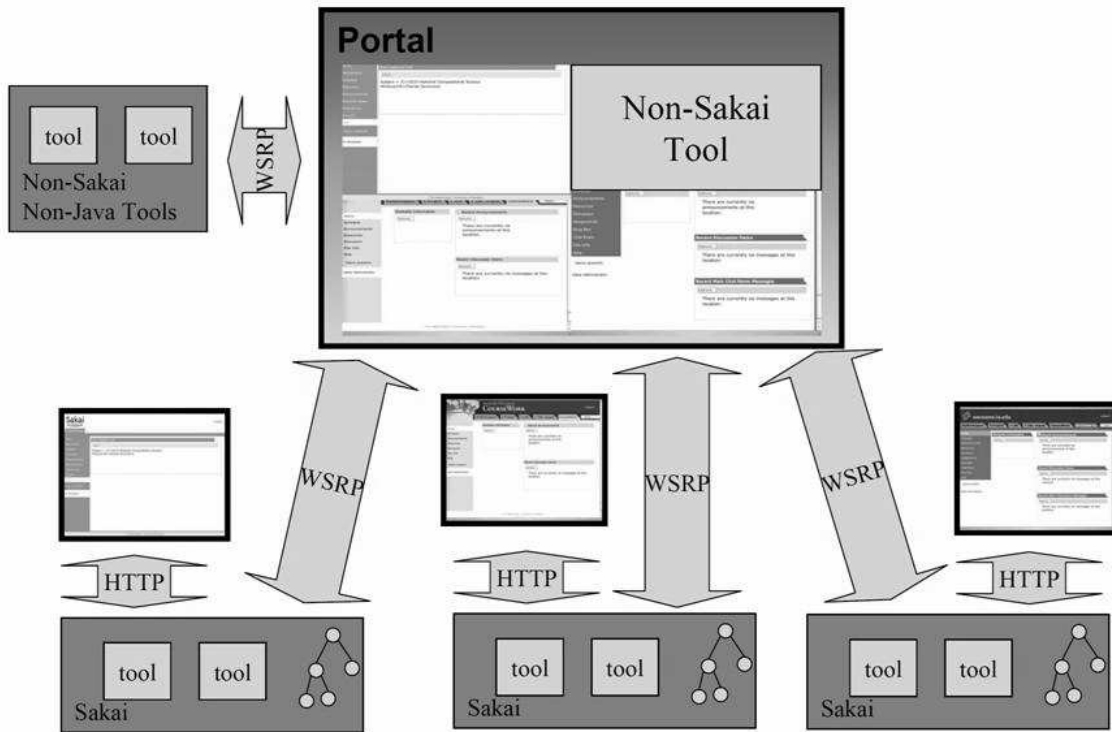
Figure 9: Pluggable portal components

Over the few months that we have been involved in discussions with JISC and the Sakai developers, in particular identifying and classifying services and debating the use of an open service architecture the picture has changed. In line with our suggestions the Sakai team are now proposing a stronger emphasis on WSRP as shown in Figure 10.

Ultimately this permits aggregation of portal content, tools and worksites managed autonomously as shown in figure 11. The fact that WSRP is used also permits portlet content to be rendered in other environments, e.g. via a Swing interface using WSRP4J. Examples of interfaces could then be built in Swing, Matlab, VTK, Iris Explorer, etc. which are all popular.

This design work was presented in the joint VRE-VLE discussion group at the JISC-CETIS Annual Conference, Oxford, 3-5/11/04. It was fed back to the conference in a plenary session by Charles Severance. Work is ongoing to establish a shared framework with an agreed set of service names and def-

The “New” Big Picture (9/04) *



* There may be a surprise or two along this path as well.

Figure 10: New Big Picture

Integration Architecture

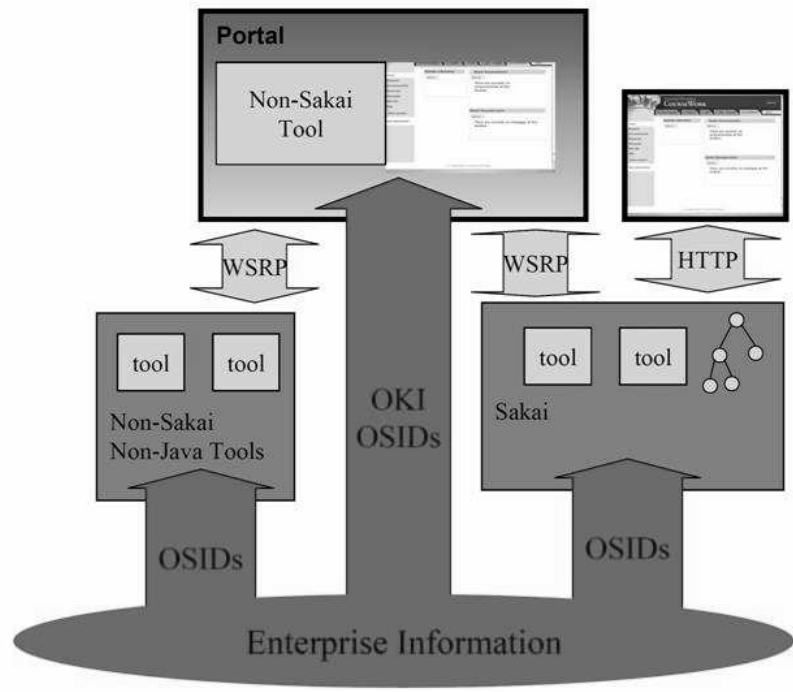


Figure 11: Portal Integration Architecture

initions for the JISC areas and to develop the technologies to deliver them to end users. See <http://www.grids.ac.uk/ETF/public/WebServices/classes.html> and below.

Appropriateness of Sakai in a VRE

We here note some comments from the commercial sector about portal deployment and usage. Key to this is a consideration of Return On Investment (ROI). Several reviews have indicated that this is a critical business factor in adopting portal technology as access to services and their provision can then be consolidated.

According to Gartner in 2003 [89], portal technology will become a key component of software suites. While Gartner identifies two types of suites, an article by C. White [88] suggests there will be three types, see Figure 12:



Figure 12: The Evolution of the Enterprise Portal – Software Suites

Intelligent Business Suite – This suite is a packaged solution that integrates the key features of an independent portal (categorization, search and personalization) with business intelligence tools, collaboration services and a content management system. This suite places emphasis on out-of-the box solutions for information access and sharing, content management and collaboration. It is similar to what Gartner calls a Smart Enterprise Suite (SES). For some organizations, a key requirement for such a suite would be that it can run on and exploit the integration features of an application server suite in areas such as application integration and Web services.

Application Package Suite – This suite provides an out-of-the box solution that integrates an application vendor’s operational application and business intelligence packages into a portal environment. The suite also provides collaboration services and an integration bus that enables third-party product integration. This suite puts a strong emphasis on pre-packaged solutions. If the application vendor also markets an application server suite, then it is likely that the application package suite will be developed and integrated with that product.

Application Server Suite – This suite brings together the four key integration infrastructure technologies (i.e. user interface, business process, application and data) in a single package that is combined with an application server and collaboration services. This suite places emphasis on application integration and on an infrastructure for building an integrated business environment. It is similar to what Gartner calls an Application Platform Suite (APS).

According to White, portal technology will become a key component of other software solutions. In developing a portal strategy, the needs of the business should be the primary focus; but it is also important to take into account other IT strategies in areas such as business intelligence, content management, collaboration and application integration. New and evolving technologies such as XML and Web services are also likely to play key roles as portals evolve into software suites.

These statements clearly reflect our investigations into the use of portals in a Virtual Research Environment. In particular the ability to combine application suites in a single portal environment, now using portlets and an additional integration API as shown in Figure 9, mirror the need to combine services from e-Learning, Digital Information and e-Research.

Portals have for a long time also been identified as a focus for delivery of a Virtual Learning Environment. An interview with Chuck Severance discussing the similarities and differences between the approach adopted by Sakai and the JISC Framework Programme is recorded on the CETIS Web site: <http://www.cetis.ac.uk/content2/20040503155445>. This includes a high-level comparison created by Mark Norton in discussion with CETIS staff, Figure 13.

The HIVE approach to e-Research presented here can be applied in many other contexts. In e-Learning we could for example use it to construct a Grid of distributed content that could be aggregated in the ways required dynamically by the user for each learning situation they face. The content of each HIVE server could be watermarked to identify its origin. This use of the HIVE would require the development of new tools, e.g. cross searching tools. Eventually we could use the HIVE to coalesce the appropriate combinations of information, e-Learning, e-Research, e-Collaboration, e-Management, e-Authoring and e-Publishing, e-Leisure tools as required by our current activity.

Services in other Frameworks

Clearly a VRE is more than just a portal! We believe that portal and portlet technology has a major role to play in delivering VRE services and tools, particularly the collaboration tools, to end users, but existing applications, e.g. GUIs must also access VRE functionality. In order to do this a lightweight toolkit is required to be downloaded onto workstations and PCs which will avoid the “client problem” of bloated and difficult software installations and associated firewall issues.

A need has been identified within the e-Science community for a client toolkit which can provide very light-weight but extensible access to Grid resources. GROWL: *Grid Resources on Workstation Library* <http://www.grid.ac.uk/GROWL> is a prototype library which could be used for this evaluation work. We are initially creating libraries in C/ C++ and R, interfacing to a set of existing services derived from HPCPortal, DataPortal and InfoPortal which are part of IeSE. It should be possible to install this library on a variety of client workstations with a minimum of additional software. The library is targetted at existing applications in physics, chemistry and statistics.

Presentation of services through both portlets into the portal framework and also as a programming library can be achieved using language-agnostic Web service client interfaces to the VRE exposed in a service oriented architecture. Early work with GROWL at Daresbury and Lancaster has shown that this is feasible in C and R languages in addition to Java. “Heritage” applications can in this way be Grid enabled or themselves exposed (wrapped) as remote services. Other examples exist, such as GEMLCA from University of Westminster [75], WSRF::Lite from RealityGrid [76] and gLite from EGEE [77]. These are responses to the recent discussion of lightweight toolkits by existing and potential Grid users. They are also crucial steps in bringing specialist applications into play for a wider research community and for achieving inter-disciplinary research agendas.

Back-end Resources and Services required in a VRE

A lot of effort in this project has gone into considering the integration of existing resources and services through the architecture proposed here. This proposal has emerged alongside very similar ones for the JISC VLE and IE programmes. All teams are in contact with the same American and European developers. A list of UK resources which *should* be integrated into the VRE were included in paper [45] and are repeated in Appendix G. Similarly the services which we believe *should* be available via a VRE are listed in Appedix H and a fuller version is being kept up to date on the ETF Web site <http://www.grid.ac.uk/ETF/public/WebServices/classes.html>. Interestingly the Integrative Biology project <http://www.integrativebiology.ac.uk> has identified a very similar set among its project deliv-

enables to enable a new insight into how chemical and biological phenomena affect the behaviour of whole organs [78].

It is important for the tools in the VRE to be able to access all the underlying resources that a particular group of collaborating researchers would require. These include computer systems, databases, data/ information collections, application codes and instruments for on-line observation and data recording and annotation. Some of the resources to which the tools that we are developing will provide immediate access include:

- HPCPortal and InfoPortal functionality applied to NGS, JCSR Grid clusters and researchers' own facilities;
- Network Monitoring tools;
- DataPortal cross search tools for scientific data;
- CCLRC's Atlas Data Store;
- HPCx capability compute facility;
- Sample experimental facilities on Daresbury Synchrotron Radiation Source;
- NGS Nodes;
- ReDReSS training and awareness content and services.

Indirect access would be provided to:

- RDN data via SPP cross search tools;
- MIMAS, Manchester Information and Associated Services;
- CSAR computing and data facilities.

We would also seek to bring in other resources and services over a longer period. Further information is provided in Appendix G.

D.1 Web Services and VRE Extensions to Sakai Framework

In discussion with the Sakai development team, we have identified four generic areas for a demonstrator VRE project to enable a wide selection of tools to be integrated as discussed. Framework extensions would be made to accommodate emerging authentication and authorisation systems such as Shibboleth and PERMIS and SOAP-based interaction with remote services such as WS-I Web services and WSRF Grid services as well as peer-to-peer services. These will be included as an "Integration API" which could eventually extend the OKI OSIDs. These extensions are focussed on enabling a portal, such as Sakai, to work in a distributed Service Oriented Architecture. This could be by accessing remote Grid or other services, aggregating content, or enabling communication between portals in a way which differs from the usual client-server model, e.g. peer-to-peer. This breaks the potential bottleneck associated with the usual monolithic portal interface (although in practice that has not been seen to be a problem with thousands of users of institutional portals based on CHEF).

In the evaluation we have attempted to investigate the design of a generic JSF interface to Web services via WSDL and a UDDI registry.

Identification, or specification, of an XML grammar for describing of collaborative research. This work is concerned with the problem of describing a collaborative research session in a standard, easily machine parsable fashion. Firstly we need to be able to describe the time of the proposed session. We then need to be able to describe its subject matter in as rich a set of terms as possible. This contextual description will be utilised in other tools, e.g. for information retrieval. Finally, we need to be able to describe the participants. The *eduPerson* initiative <http://www.educause.edu/eduperson/> will be assessed as a suitable source for a descriptive grammar. The *eduPerson* initiative is aimed at providing a standard way for institutions to list individuals in their LDAP directories. One of the outputs of the *eduPerson* project will be a controlled vocabulary that could be used in an XML grammar such as the one proposed here. Some background work has been done by the project partners to extend the UDDI service registry schema to describe projects, people, resources and applications <http://www.grid.ac.uk/Papers/Schema/schema.pdf>.

Other input will be taken from the CCF, Collaborative Computing Frameworks joint project between University of Reading and Emory, USA.

A Service Authentication and Identity Verification System. A concern with establishing virtual collaborations, is one of identity. How can you be reasonably sure that the colleague you are working with 500 miles away is the person they say they are? Any Grid tools accessed via Web services are likely to require Grid Security Infrastructure (GSI) type or other appropriate authentication. This work package will implement a Shibboleth Federation (see <http://shibboleth.internet2.edu/>) consisting of the collaborators' institutions. Access to the VRE will be protected by a resource manager, which will delegate authentication to the callee's institution. The keys exchanged during the Shibboleth authentication phase will then be used for generation of the appropriate Grid credentials required for use of the Grid tools contained within the VRE. This will build on the JISC evaluation in which Sakai has already been linked to an institutional LDAP people system. Other technology could be harvested from HPCPortal or OGCE which use x.509 extension certificates and a MyProxy certificate repository hosted at the UK Grid Support Centre <http://myproxy.grid-support.ac.uk>.

Another concern is with establishing appropriate authorisation mechanisms respecting local policies. PERMIS provides a system using a role-based set of authorisation policies. It is the job of the tool provider to outline the privileges of the different roles and to the remote site to assign roles to its users. We will seek input from JISC-funded Security Middleware projects on this and related matters. A separate bid is being prepared to enhance the management interface for PERMIS in these areas for use with Sakai by the Salford group.

A JSF-based Web service interface generator. If we wish to provide access to Web services as tools within the Sakai framework, we need to provide a user interface for parameter input. Within Sakai such interfaces are rendered using a pipeline consisting of an abstract XML layout description and a final Java Server Faces user interface. This work package will produce code that generates such an interface from the WSDL file of a desired Web service. All that will be required is the URL of the service WSDL. We will seek input from the Indiana Xportlets group who have done related work `xportlets`.

Service registries are a key technology enabling shared development, code distribution and re-use and version management. This functionality will be capable of being coupled with UDDI registry lookups so that a VRE user will be able to search for appropriate Web services to integrate as tools, select the most desirable ones and have user interfaces transparently created there and then. Prototype UDDI servers are being hosted at Daresbury and Oxford. Background work is reported in <http://www.grid.ac.uk/Papers/UDDI/uddi.pdf>.

This work will have to be closely coupled with the identity work as the credentials gathered at Sakai logon will have to be passed onto any Web services requiring authentication. This is done in HPCPortal and GROWL using agents, a session key and MyProxy and similarly in the GT3.9.1 implementation of WSRF.

Interface to Peer to Peer Tools This task is to provide generic interfaces to integrate tools using peer-to-peer technology such as JXTA in a straightforward way. Its scope is largely still to be defined, but it is a response to the emergence of peer-to-peer tools such as LionShare which would ideally be accessible in a VRE through a single interface. The interface will also support the peer-to-peer communication between portlets in multiple portal instances (another form of aggregation). A complementary proposal to use P2P services to establish a VRE is being submitted by the Reading and Westminster groups. Background work is reported in <http://www.grid.ac.uk/Papers/Rana/rana.pdf>.

D.2 Implementation

We instantiated a Web service client portlet to access some of the Grid tools in WP3.

D.3 Technical Specification

This is ongoing now as part of the Sakai VRE Demonstrator project.

E Another independent Technical Report

This report from an unknown source had been circulated to a few people and was obtained by us under an agreement that it would not be circulated further. Since a number of decisions may have been taken subjectively based on its content we however consider it defensible to publish the text here with our comments on the factual content in our objective evaluation of Sakai.

SAKAI Technical Report

August 2004

Introduction

This report is as an objective view of the publicly available download of SAKAI Release Candidate 2.

The goals of the SAKAI project are to produce the following:

- Specification for a framework allowing plug & play components to be deployed
- Developer toolkit for new SAKAI friendly tools to run in the framework
- Reference download to demonstrate the concepts and for anyone to use under license.

The scope of this paper is not evaluate how the available code has met these objectives since the web site is clear that they will not be met until SAKAI version 2.

The current distribution is essentially an updated version of the CHEF tool being exposed through the uPortal content aggregation system. CHEF was developed at the University of Michigan as a collaborative tool for teaching. Future releases will include as assessment system called Samigo and a content delivery mechanism replace OnCourse, and Open Courseware.

Architecture

The architecture assessment is made from the existing source code made available from the SAKAI project site and documentation from the SEPP meeting in July.

The Tools Portability Profile (TPP) is a specification for constructing end user facing components that can plug directly into any SAKAI compatible framework. The specification is a recommended use of the core technologies (see below) and a package structure for deploying components. The specification is not sufficiently detailed

or stable to ensure true portability across implementations at this stage. This concern was recently highlighted in the re-prioritization of WSRP instead of JSR168 for portlet aggregation.

The biggest concern for the current code structure is the lack of a domain model. The current code and documentation for plugging in components is neutral of any domain or information model. As a result, the current release looks like yet another development framework.

The concerns over the domain model may be addressed with the introduction of the Samigo assessment model as the first education related component to be used in the SAKAI framework. It is hoped the design team demonstrate good practice in separating the domain model from the implementation dependencies.

The current distribution file system layout does not reflect the package naming structure and as a result is somewhat confusing. The purpose for the layout is not clear other than supporting a distributed development team where new components are simply created at the file root level. This approach makes it very difficult to see how a shared information model will be supported.

There is significant code duplication in the current distribution but that is most likely a result of the combination of two major projects in a short time scale and will be addressed in later releases.

The reference code looks to have been unit tested without the use of JUnit as a testing framework. As well as being best practice, the inclusion of a JUnit test harness structure would better support the development of components for use in the framework providing a consistent way to test for interoperability. This observation is made from the numerous places where a main function used for testing has been commented out.

Core Technologies

The core technologies of the framework are a collection of widely accepted best practices for java based web applications:

The component architecture is based on the Spring Framework (www.springframework.com) which is a widely recognized framework for developing web applications without a full Java 2 Enterprise Edition (J2EE) managed container. The downside is that the Spring Framework is only recommended over J2EE when the full systemic qualities and service levels of enterprise software are not needed.

The presentation layer utilizes Java Server Faces (JSF) for implementing the Model View Controller (MVC) design pattern and ensuring the presentation and user interaction is abstracted from any application business rules.

Configuration management is driven by the Apache Maven tool for build and deployment. The scripts are available with the reference download and are written for deployment into a Tomcat container with an

in-memory XML database or Oracle setup.

Conclusions

The project is still in very early stages and the team is focused on delivering a working solution for the Universities of Michigan and Indiana. It could be argued that the SEPP initiative should have been delayed until the architecture and design was stabilized given the short-term focus on a deliverable system.

The problem is that these are really two separate projects attempting to share a single constrained resource and not necessarily having the same vision. Without a formal mechanism for collecting requirements from the SEPP members and allowing the architecture team time to refactor the design, the program will never realize it's vision of a truly open shared platform.

Some comments on this report follow.

The Tools Portability Profile (TPP) is a specification for constructing end user facing components that can plug directly into any SAKAI compatible framework. The specification is a recommended use of the core technologies (see below) and a package structure for deploying components. The specification is not sufficiently detailed or stable to ensure true portability across implementations at this stage. This concern was recently highlighted in the re-prioritization of WSRP instead of JSR168 for portlet aggregation.

- The TPP is designed to make tools portable across Sakai installations, not across other portal frameworks. It is specified in enough detail to enable this portability.
- The architectural switch to WSRP demonstrates the efficacy of the SEPP process. The switch is designed to permit the crafting of tools in other languages, and the exporting of Sakai worksites directly into WSRP consumers. For more information please refer to *sakai_architecture_evaluation.pdf*.

The biggest concern for the current code structure is the lack of a domain model. The current code and documentation for plugging in components is neutral of any domain or information model. As a result, the current release looks like yet another development framework.

- Sakai is domain agnostic. Sakai is designed to provide a stable software framework into which to plug tool components, together with a useful set of collaboration tools pre-plugged. It is NOT a development framework in the sense of JSF, Struts or other programming libraries, it is more of a production line framework - just plug your tools in and customize for your institution. The domain specific elements, and the required models, come with the tools.

There is significant code duplication in the current distribution but that is most likely a result of the combination of two major projects in a short time scale and will be addressed in later releases.

- There is a significant amount of FUNCTIONALITY duplication in the Sakai codebase. This is because a pragmatic decision was taken to write adapters for the existing (and production tested) CHEF tools to allow them to work within the new framework, whilst coding up new, "native", versions.

The reference code looks to have been unit tested without the use of JUnit as a testing framework. As well as being best practice, the inclusion of a JUnit test harness structure would better support the development of components for use in the framework providing a consistent way to test for interoperability.

- The unit testing used in Sakai is a mixed bag. There are some JUnit tests in the legacy calendar tool code, and quite a lot of ad hoc tests within main functions. A point worth noting here is that unit testing is a well understood art and does not need JUnit to be successful.

It could be argued that the SEPP initiative should have been delayed until the architecture and design was stabilized given the short-term focus on a deliverable system.

- The SEPP could well have been delayed. This would have given advantages, as well as disadvantages. It would have meant less distractions for the core team, but could have also cut out a major source of user feedback. The SEPP serves several purposes; it provides user input, testing and development funds. Ultimately the SEPP could be seen as a useful outcome of the Sakai project as it is hoped that it will result in new models for educational collaboration on open source software development. The mechanism for requirement gathering probably does need formalising to a degree, but the foundations are there. There is a dedicated team for eliciting user requirements and producing GUI wireframes for the software team; this is probably more than many software houses have.
- Sakai and CHEF are not two separate projects, what we are seeing is the gradual transformation of CHEF into Sakai, as stated on the Web site.
- A final point to make is that there are several effective players in the VLE market. If Sakai delivers on its promises and is marketed successfully, it will take considerable market share from other VLE companies like Sun, Blackboard and WebCT.

F Integration of UK VRE Tools into Sakai

In this section we describe a number of tools which have been discussed during the evaluation and which we believe would form useful components of a Sakai-based VRE. Some of these will be developed by porting open-source software under the Sakai VRE Demonstrator project funded by the JISC programme http://www.jisc.ac.uk/index.cfm?name=programme_vre.

CopperCore (Lancaster) Learning Design content player for use in training and awareness applications.

IRC Chat Tool. (Lancaster) IRC is a widely used protocol for on-line chat sessions including group or 1 on 1. Users can choose between several threads. A Chat portlet is already present in the Sakai framework but work is needed to integrate chat sessions from project worksites into a user's personal worksite and to link to external IRC channels. Jabber is an alternative.

Blog. (Daresbury) A candidate exists, Blojsom at <http://www.blojsom.com>.

Programmable Scientific Calculator. (Daresbury)

Active Spreadsheet. (Daresbury)

Personal Address Book.

Personal Information Guide (PIG). Office Assistant and help tool in the form of a flying pig.

Helpdesk Interface

Distributed Word Processor. A distributed word processing tool to facilitate collaborative document editing.

Shopping Cart. (Daresbury) Adaptation of the personal resource tool designed to package data and executable programs for processing, e.g. via the Grid.

Distributed Presentation Tool. A distributed presentation tool for multicasting slide shows to co-researchers.

Wiki. (Daresbury) Java-based portlet interface to popular collaborative Web editing tool. A prototype exists from Michael Fischer's computational anthropology group, University of Kent. Another candidate is Xwiki.

Research Scheduling Tool. (Lancaster/ Daresbury) This package will involve the development of software adapters intended to query potential co-researcher's Exchange (or any iCalendar compliant mail server) calendars. A software module will be produced designed to query potential co-researcher's calendars and list potential research time slots.

Cross Search Tool. (Oxford) An appropriate tool for cross searching digital resource collections is available as a portlet from the Subject Portal Project. This currently searches the RDN, Resource Discovery Network, collections.

Relevant Information Retrieval Tool. (Lancaster/ Daresbury) This work package will focus on the delivery of a WSRP portlet based tool enabling lookup of resources relevant to the research being undertaken in the VRE at the current time. This focusses on context based content aggregation using the output of the grammar extension deliverable. It will be used in that the subject matter portion of the VRE session description to help generate the information search criteria. Each researcher involved in a project will be able to manage content and publish it for other researchers who can retrieve it from other portal instances. Recent work has shown how WSRP services can be published via UDDI. There are relevant Web-based UDDI clients which could form the basis of this tool.

Personal and Project Notebooks. (Daresbury) This will be an adaptation of an existing e-Notebook tool. Database backed "blog" style tools will be produced to enable the making of research notes.

Personal Information Manager. e.g. for sharing schedules with other software. Example, Chandler.

Project Bibliography Construction. (Daresbury) Tool to enable collaborators to upload bibliographies from their work machine into a shared project bibliography. A BibTeX tool exists in OGCE which can be adapted to produce output in different formats.

A Statistical Computing Web Service (SABRE and R). (Lancaster) Work package 2 is concerned with creating a resource producer, in the form of a web service version of the R and Sabre statistical tools. The Sabre routines will be made available via R calls, and these in turn will be made available as a multiple message web service. A prototype of this has been demonstrated in the GROWL project at Daresbury, see <http://esc.dl.ac.uk/GROWL>. This service will be able to run on the National Grid Service (JCSR nodes at Manchester, Leeds, Oxford, RAL) and will require grid authentication via Sakai's Grid logon facilities, as enhanced in the grammar. Initially, a dataset will be uploaded from a collaborator's computer to the service for processing, although the possibility of remote dataset specification will be investigated. GridFTP or SRB client calls would be used for this.

Video Conferencing and AGN Participation. (All) This work package will build on the work done by Geoffrey Fox from Indiana on the Narada Brokering based GlobalMMCS video conferencing system. It is anticipated that the bulk of the work will involve configuring GlobalMMCS, devising an elegant deployment solution and coding the conference instantiation facilities into CHEF. GlobalMMCS uses a message exchange protocol based on SOAP for setting up conferences. Communications are then carried out using message packets, which can be shunted over a self-organising network overlay of Narada Brokering nodes. A browser borne video conferencing tool tool, for use by VRE research project members will be delivered.

Scientific Data Management. (Daresbury) We will port CCLRC's DataPortal JSPs to portlets to provide cross-search access to a range of scientific data via a common metadata model. The "shopping cart" portlet will connect a list of identified datasets to a GridFTP facility or other functions.

Visualisation. (Daresbury/ RAL) Visualisation tools will be imported from other projects, for instance the e-Science Visualisation Group at CCLRC Rutherford-Appleton Laboratory and Ken Brodrie, University

of Leeds, which provides tools for the GODIVA, GViz and Integrative Biology projects and University of Manchester's SVE Group.

Distributed Whiteboard. (Lancaster/ Reading) Input will be taken from the CCF, Collaborative Computing Frameworks joint project between University of Reading and Emory, USA which includes a multicast-enabled "clearboard" and other collaboration tools. A distributed "whiteboard" tool is required, to facilitate collaborative discussion and annotation of research materials such as images and documents.

Distributed Flowchart Editor. (Lancaster) A distributed flowcharting tool to assist in the planning of research activities.

Text Editor Emacs style editor

XHTML Editor

Distributed Display. (Lancaster) A distributed display tool for multicasting live data from HPC.

Bugzilla. Collect bug reports and feedback to developers.

CVS. A Web front end for CVS exists.

TeX/ LaTeX Processor Back-end text processing and rendering linking to the BibTeX tool.

Simple Interactive Plotting. e.g. GNUplot for uploading and graphing 2D datasets.

Portal Statistics. Extension of portal group collaboration tools to support gathering of usage statistics. Similar tools exist in PHP-Nuke and are very popular.

Graphing/ charting Web service Takes a dataset and graphs it. Could use the graphing functionality of R behind the scenes.

Extended presence tool Shows co-researcher's location on a map. Hover over the point and you get metadata about that individual (like contact details).

Grid Information (Daresbury) InfoPortal tools

Network Information (Daresbury) Tools from the GridMon project are being Web-service enabled.

Grid Computing (Daresbury) HPCPortal tools.

Data and Metadata Upload and Harvesting

Project Publishing

Application Publishing (Daresbury) using IeSE XML database and search tools

UDDI (Daresbury/ Oxford) jUDDI has been implemented to test its appropriateness in a Grid context.

Portal Workspace Manager

Authorisation Policy Manager

Documentation Policy Manager

G Resources

Some resources and facilities available to the academic research community were identified during the VRE work with JCSR:

- Access Grid Nodes (e-Science Centres);
- Course Content (University and Training Institutions);
- Condor pools of workstations (University and Teaching institutions);
- Resource Discovery Network resources (JCIE) <http://www.rdn.ac.uk/>. See the RDN Internet Resource Catalogue <http://www.jisc.ac.uk/index.cfm?name=rdnipage>
- AHDS (AHRB) and e-SS (ESRC) and related training and awareness material, e.g. REDRESS;
- Directories: Z-Directory (UKOLN), Z39-50 target directory (Index Data), RSS-express (UKOLN), OAI Data providers (OAI), IESR (JISC)
- Text mining service (BBSRC), Data Curation Centre and any other specific research resources funded in partnership with Research Councils;
- Resources referenced in the JISC subject resources guides <http://www.jisc.ac.uk/index.cfm?name=resguides>. These cover the seven subject areas: Arts and Humanities; Engineering, Mathematics and Computing; Geography and the Environment; Health and Life Sciences; Hospitality, Leisure, Sport and Tourism; Physical Sciences; Social Sciences. They include resources such as: Bibliographic, reference and research information; Publications online; Subject gateways; Data services; Learning and teaching; Support services.
- Tools referenced in JISC Collections publications list: collections of high quality online research tools, learning materials and digital archives for UK HE and FE institutions <http://www.jisc.ac.uk/index.cfm?name=coll>
- National Grid Service nodes (JCSR) <http://www.ngs.ac.uk>;
- Supercomputing facilities such as HPCx, CSAR (managed by EPSRC): <http://www.hpcx.ac.uk> and <http://www.csar.ac.uk>;
- Data Archive and MIMAS (ESRC);
- Protein Data Bank (Hosted by Wellcome Foundation at EBI);
- Large-scale facilities such as SRS, ISIS, Diamond (hosted at CCLRC) and associated scientific data collections;
- LHC Data Grid (PPARC);
- NERC Data Centres and CEH;
- Telescopes, e.g. via eSTAR services (PPARC);
- British Library, National Museums, etc. e.g. <http://www.nmsi.ac.uk/>
- Others such as British Geological Survey, UK Met. Office, Hadley Centre.

JISC
Function
Mappings –
Starting
points for
coordination
within
Sakai

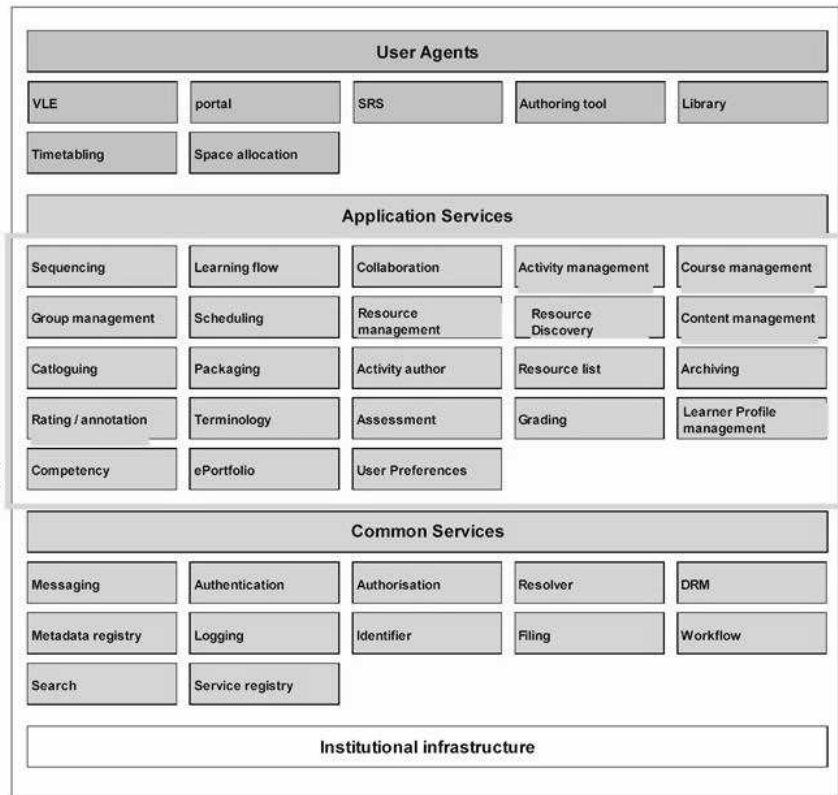


Figure 7: model of services demonstrating common and application services together with selected user agents

Figure 13: JISC Framework 1

Cross Domain Modelling and Common Services

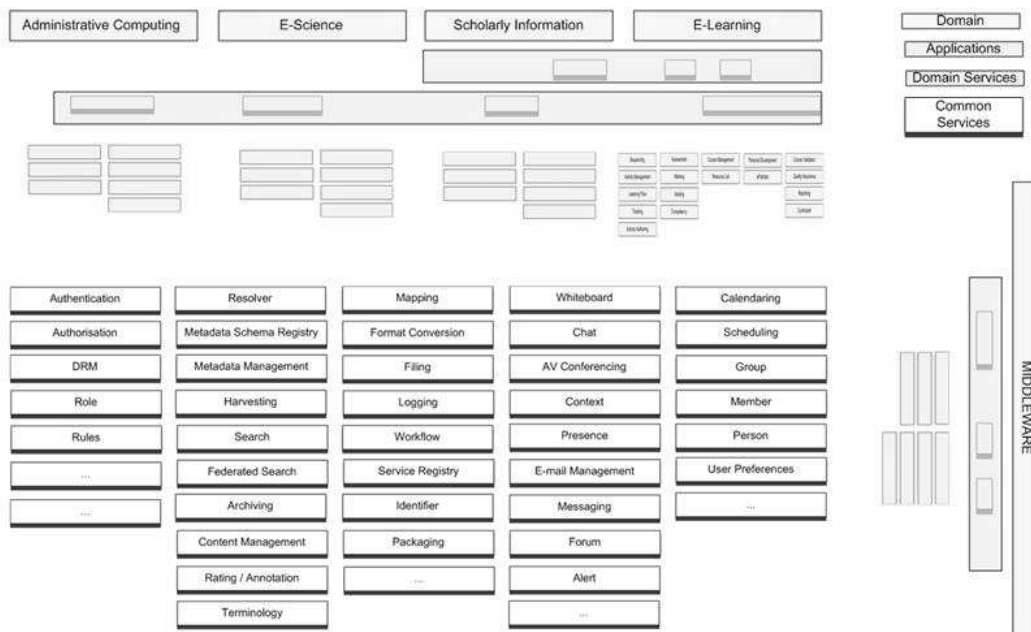


Figure 14: JISC Framework 2

H Service Capability Set

This appendix summarises ongoing e-Service classification work which will be extended online at <http://www.grid.ac.uk/ETF/public/WebServices/classes.html>.

We list services which could form the basis for virtual environments for a variety of purposes. We specifically draw upon considerations of an information environment; virtual learning; and e-research. Reports and papers from which ideas have been taken are listed in the references. We thank the authors of these and also groups such as JCLT, JCIE and ETF whose members have been debating e-services for at least a year. Additional input has been taken from the various working and research groups of the Global Grid Forum which is currently identifying services, specifications and standards leading to an Open Grid Services Architecture (OGSA). In the UK, the recommendations of the Grid Architecture Task Force and the e-Science Gap Analysis carried out by Geoffrey Fox and David Walker have been taken on board.

We have attempted a rather broad and arbitrary classification of the services identified into the following areas: collaboration; e-research; e-learning; digital information; common infrastructure.

We do not consider these to be definitive lists of the services that can be provided, only examples, and we hope that additional services will be identified and developed in ensuing programmes, or identified services refined in the light of future requirements analyses ? such lists are organic and will grow and shrink. The aim is to kick-start a programme whereby a framework can be deployed to enable community input and contribution of more specialised services and resources. There is a tendency at the start to list many small-sized atomic services for every function imaginable, later the need to optimise the large-scale distributed system may indicate that services have to be aggregated (federated) in different ways to improve performance. We expect any e-Environment to support only a range of the services listed here.

e-Collaboration application services

Collaboration is about people working together, either as peers or in some more formally-defined relationship, such as researcher-supervisor. Collaboration extends to high-end technologies such as Access Grid.

- Calendar
- Collaboration Management
- Content Management
- Content Sharing
- Group or VO Management
- Peer Group Join
- Peer Service Location
- User Registration

e-Research application services

Research services are specific to supporting research processes and tools, including the Grid. They should include collaboration with experts and peers, encapsulation of complex procedures for non experts to facilitate growth of inter-disciplinary sciences and aids for results publication and proposal writing. We aim for a holistic system-based approach!

- Application Management
- Deployment
- Distribution
- Fabric Management
- Grid Information
- Information Access
- Information Aggregation
- Information Content Registration

- Information Query
- Information Metadata
- Information Presentation
- Information Notification
- Information Update
- Job Management
- Knowledge Extraction
- Knowledge Syndication (Join)
- Process Building
- Proposal Writing
- Resource Discovery
- Resource Management
- Scheduling
- Security
- Validation and Verification
- Visualisation

e-Learning application services

Learning services are about supporting a Managed Learning Environment with particular relevance to teachers and students supporting both peer groups and training hierarchies. A variety of approaches to teaching can be included with also self learning and assessment. A training and awareness environment might be distinguished from a more formal teaching environment because the former may not require assessment but could have more interactive demonstration material (e.g. via the Grid). There should be the ability to walk through material in various ways, log activities and attention to accessibility issues. Again the aim is to provide a holistic approach, but there must be access to humans if a student runs into difficulties, be they ones of understanding or personal. An electronic system cannot completely replace tutors!

- Activity Authoring
- Activity Management
- Assessment
- Competency
- Course Management
- e-Portfolio
- Grading
- Help
- Learner Profile Management
- Learning Flow
- Rating/Annotation
- Resource List
- Resource Management
- Scheduling
- Sequencing
- Trails and Personalisation
- View

Digital information services

These are services for digital information and data management, which will be informed by issues identified by the Digital Curation entre. Special attention to database servers and large collections, some of which are of qualitative nature, will be required. Conversion to electronic format and data mining may be targeted too.d

- Archiving

- Cataloguing/Curation
- Data Access and Integration
- Data Virtualisation
- Data Replication
- Data Management
- Deposition
- Dictionaries and Ontologies
- Digital Rights Management (DRM)
- Resolver Services
- Resource Discovery
- Terminology

Common services

We have identified the following common services which may underpin many of the other services. They could in many cases be provided as part of the "infrastructure", and not directly visible to the users, maybe acting as "agents" to other high-level services. Session management is an example of this. Common services also give scope for system optimisation, e.g. through aggregation or federation in special cases and could be encapsulated in workflow for various scenarios. Some use cases are required here to extend and refine the list.

- Accounting
- Alert/Notification
- Authentication
- Authorization
- Billing
- Component Communication
- Filing
- File/Dataset/BLOB Management
- Identifier
- Logging
- Messaging
- Metadata Registry
- Monitoring
- Network Management
- Packaging
- Personalisation
- Portal Services
- Proxy Management
- Search
- Service Registry
- Transaction
- User Preferences
- Workflow

I Generic Portal Engines

Tables I and I list a number of generic portal engines and some related software from commercial vendors and public-domain developers. These were identified in the *Portals and Portlets 2003* workshop [28] and a subsequent technical survey by Adrian Fish for the ReDRESS project [37]. Other links have been added over the course of this evaluation.

Table 3: 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
Chandler	http://www.osafoundation.org/
CMIS ePortal	www.ccmsoftware.com/cmisis.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/portal/
Oracle Portal	http://portalcenter.oracle.com
RedHat Portal Server	http://www.redhat.com/software/rha/portalserver/
SITE e:Vision	http://www.sits.co.uk/
Sun iPlanet Portal Server	http://www.iplanet.com/products/iplanet_portal
SunGard	
Terminal 4	
Unicon Academus	http://www.uicon.net/academus
VKP	http://vkp.leeds.ac.uk/
WebCT Vista	http://www.webct.com/products/viewpage?name=products_vista

Table 4: Generic Portal Engines 2

	Public Domain
Bodington	http://www.bodington.org
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
Groove	http://www.groove.net/
JetSpeed	http://portals.apache.org/jetspeed-2
jPortlet	http://jportlet.sourceforge.net/
LifeRay	http://www.liferay.com/products/index.jsp
LionShare	http://lionshare.its.psu.edu/main/
MyLibrary	http://www.lib.ncsu.edu/eresources/mylibrary/
OGCE	http://www.ogce.org
PHP-Nuke	http://phpnuke.org
PostNuke	http://www.postnuke.com
Privaria	http://www.privaria.org/
Sakai	http://www.sakaiproject.org
StringBeans	stringbeans
Thatware	http://thatware.org
uPortal	http://mis105.mis.udel.edu/ja-sig/uportal
Zope	http://www.zope.org/