

A Web Portal for the National Grid Service

Xiaobo Yang, Dharmesh Chohan, Xiao Dong Wang and **Rob Allan**

CCLRC e-Science Centre, CCLRC Daresbury Laboratory
Warrington WA4 4AD, UK

Abstract

This paper gives a description of the portal user interface under development for the UK National Grid Service (NGS) by the Grid Technology Group at CCLRC Daresbury Laboratory. The first release of the NGS Portal was based on CHEF, the *CompreHensive collaborativE Framework* originally developed at the University of Michigan. A set of Grid-related tools has been successfully deployed to provide transparent access to NGS service through a familiar Web interface. A CHEF internal service has also been developed to realise Single Sign-On (SSO) via the MyProxy server maintained by the UK Grid Operations and Support Centre. The second release of the NGS Portal is the described which is based on StringBeans, a JSR 168 portal framework. StringBeans has been customised to enable MyProxy login for Grid users through JAAS (Java Authentication and Authorisation Service) by using the MyProxy Login Module. The NGS portal is now being “cloned” for other projects to benefit from the re-usable Java portlet technology.

1. Introduction and Background

The National Grid Service (NGS) [1], funded by JISC, CCLRC and EPSRC, is the core UK Grid, which has been deployed for the production use of computational and data Grid resources in all branches of academic research. Currently the NGS core Grid consists of four clusters - two compute clusters and two data clusters, plus two national HPC services. The four clusters are located at the University of Manchester, the CCLRC Rutherford Appleton Laboratory, the University of Leeds and the University of Oxford. The two national HPC services are Manchester based UK National HPC Service (CSAR) and the CCLRC Daresbury Laboratory based HPCx. Innately these resources are geographically distributed and provide an ideal test bed for Grid techniques. Two additional nodes from the University of Bristol and the University of Cardiff have recently been connected to NGS. Overall the NGS provides access to over 2,000 processors, and over 36 TB of “data grid” capacity. In order to provide users with transparent and easy access to these resources, the NGS Portal has been developed and deployed for production usage.

Many Web based portals have been set up for distributed Grid environment to improve the usability of Grid, for example, the Telescience Portal [2], the HotPage Grid Computing Portal [3] and the Cambridge CFD Grid Portal [4]. Work in the Grid Technology Group at CCLRC Daresbury Laboratory has focussed on integrating Grid services and collaboration tools into a familiar Web portal environment. This is

in the context of the *Integrated e-Science Environment* [5]. Early versions of such portals [6] were developed in conjunction with the Level-2 Grid deployed by the UK Grid Engineering Task Force.

In this paper, we will first give a brief introduction of the prototype portal and feedback from users. Then two releases of the NGS Portal, release 1.0 based on CHEF (a *CompreHensive collaborativE Framework*) and release 2.0 based on StringBeans (a JSR 168 compliant portal framework) are followed. At the end of the paper, conclusions will be drawn with our plan of future work.

2. Prototype Portal and Feedback from Users

An international workshop was held at NeSC 14-17th July 2003 [7], shortly after which the WSRP [8] and JSR 168 [9] standards were ratified. Work of the SDSC and Indiana groups from the USA were presented at this workshop along with many European projects. The Open Grid Computing Environment portal was announced as an NSF-backed project for the US National Middleware Initiative. This is based on CHEF, a comprehensive collaborative framework originally developed at the University of Michigan (CHEF is now replaced by Sakai [10], a project for building and deploying a new Collaboration and Learning Environment (CLE) for higher education), which had been extended for Grid use principally in the NEESGrid project [11].

Based on this architecture and its functionality, CHEF was selected as our basis

for the first release of the NGS Portal. CHEF was developed as an on-line service that supports working communities. The basic idea is to provide geographically-separated groups of students, teachers and classes with a consistent interface to work together through a familiar Web browser. As a virtual learning environment, CHEF provides a set of collaboration tools such as announcements, discussions, resource page, email, chat, instant messaging and on-line presentation. In order to provide users easy access to NGS computing power, some Grid tools have been developed and added to this set based on Java CoG [12].

Whilst user feedback has been generally favourable, some suggestions were made to enhance the look and feel and usability of the portal particularly with respect to Single Sign-On (SSO) and on-line help. SSO is considered to be an issue by all Grid portal developers worldwide. In developing our NGS Portal, SSO is currently realised by using an external secure MyProxy [13] server.

3. NGS Portal Release 1.0

3.1 Tools Developed for Accessing NGS Resources

Based on CHEF, a set of Grid-related tools listed below were developed for the first release of the NGS Portal in 2004.

- MyProxy management
- GRAM job submission
- FileTransfer based on GridFTP
- MDS resource discovery

Through a Web interface, the above tools provide the transparent access to NGS resources. The MyProxy management tool is used to retrieve/renew grid proxy credential for end users through the NGS MyProxy server. While the job submission and file transfer tools realise the main Grid-related functionalities for submitting a job to a remote computing resource and managing file transfer between local computer and a remote NGS node or between two NGS nodes (3rd party transfer). A screenshot of the FileTransfer tool is give in Fig. 1.

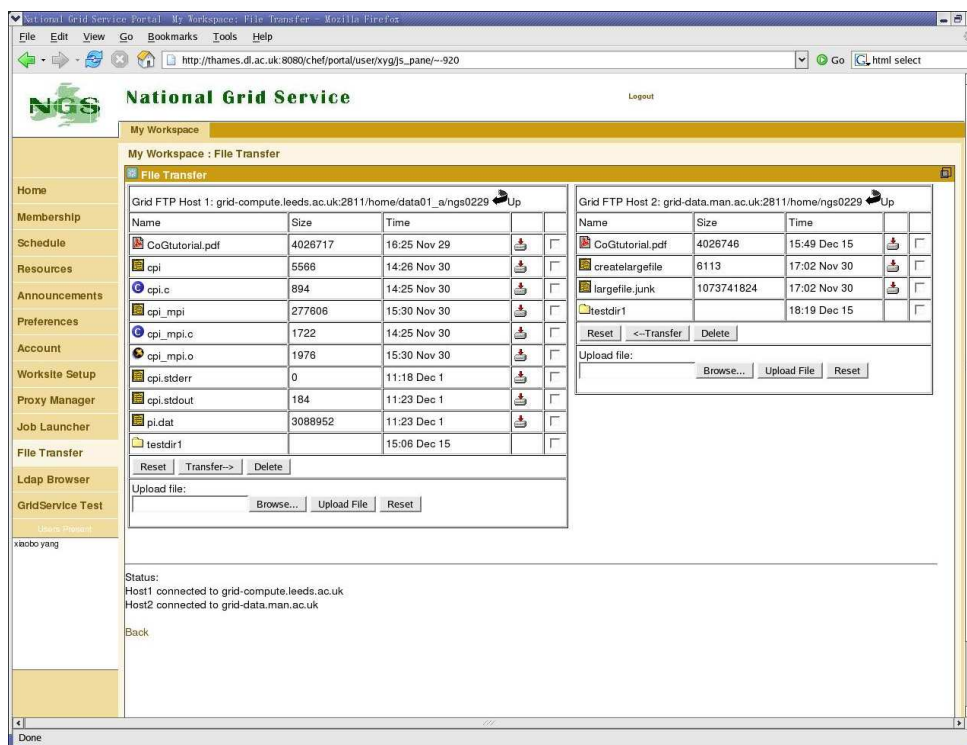


Figure 1: A screenshot of FileTransfer tool in the NGS Portal release 1.0

3.2 Single Sign-On (SSO) through MyProxy

The Globus Toolkit uses PKI-based Grid Security Infrastructure (GSI) for Single Sign-On

in a Grid environment. It is necessary to retrieve a Grid proxy credential before accessing any Grid resource. A Web-based portal which supports the full Grid functionalities usually

follows two steps to retrieve a proxy credential: 1) a user logs in the portal using his/ her login details; 2) once authenticated, the user uses a tool within the portal to retrieve a Grid proxy credential commonly through an external MyProxy server. With this proxy credential, it is then possible to access remote resources. In this way, as described in Fig. 2, the user has to remember two separate accounts, i.e., a portal account and a MyProxy account.

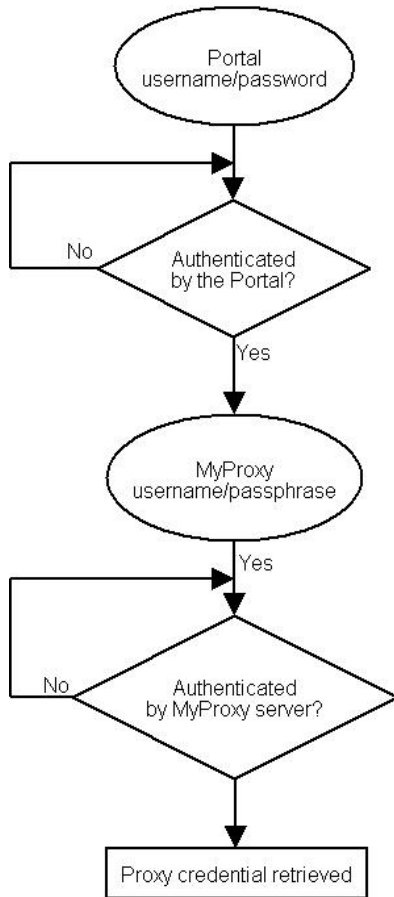


Figure 2: Commonly used two-step portal login diagram for accessing Grid resources

Since it is necessary to retrieve a Grid proxy credential from the MyProxy server for accessing the UK core Grid, a natural solution to simplify the two-step authentication is to authenticate a user through the MyProxy server. As illustrated in Fig. 3, the one-step login diagram simply authenticates a user using the MyProxy server without a portal authentication check. In the NGS Portal release 1.0, a CHEF service was developed to realise this. Once a user is authenticated through the MyProxy, the user will be mapped to a temporary account which only resides in the memory.

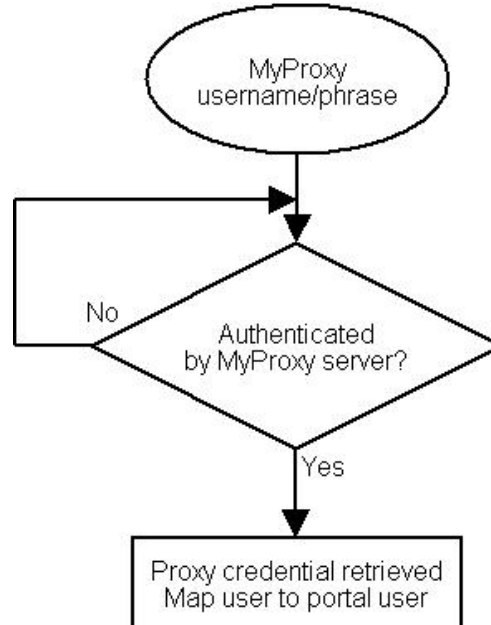


Figure 3: One-step portal login diagram for accessing grid resources

4. NGS Portal Release 2.0

4.1 Adopting the JSR 168 Standard

It became clear in early 2005 that the Sakai project was not going to be fully JSR 168 standard compliant for the foreseeable future. With pressure to deploy a fully-functional portal which could integrate tools coming from other e-Science projects and provide generic tools back for them to re-use, it was decided that a JSR 168 compliant framework was needed. Several such frameworks have been evaluated, including: uPortal, eXo Platform, Liferay, GridSphere and StringBeans, see [14]. Portlets using the JSR 168 API have been shown to be easily portable between all the frameworks tested. Because of its ease of use, configurability, and support offered by the developers, we are currently using the StringBeans portal [15] release 2.4. A screenshot of the home page of the NGS Portal release 2.0 is shown in Fig. 4.

4.2 Portlets Developed for the NGS Portal Release 2.0

With all tools developed in the NGS Portal release 1.0 converted to JSR 168 portlets, some additional portlets have also been developed such as a BatchJobMonitor portlet and a SRB portlet (not included in the NGS Portal release

2.0, but which will be added soon). All these portlets are included within a single portlet application called the Grid portlet application. The reason for including all Grid related portlets in a single Web Archive file is that according to the JSR 168 standard, different portlet applications cannot share sessions between each other. In our Grid portlet application, all portlets included can share sessions between each other as they all reside in one portlet application. This makes it possible to share Grid proxy credential through sessions between all portlets included.

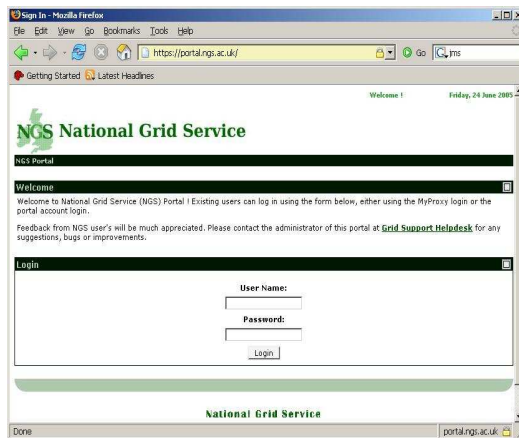


Figure 4: A screenshot of the welcome page of the NGS Portal release 2.0

According to the JAAS architecture, different kinds of authentication technologies can be plugged under the StringBeans portal framework we are using. In fact it is even possible to share information between different login modules, which is the key feature to achieve Single Sign-On among Web applications. In our scenario, one independent authentication module, MyProxy Login Module, interacts with a callback handler to retrieve username and password for MyProxy authentication. As illustrated in the first release of the NGS Portal, the one-step authentication mechanism brings the benefit of automatically retrieving end users' proxy credentials that are necessary to execute any Grid-related tasks such as submitting a job to a remote computing resource, executing file transfer between local and remote computers. Once the users are authenticated through the MyProxy Login Module, they are mapped to portal accounts

automatically. The portal accounts are therefore completely hidden from the view of end users. This procedure is illustrated in Fig. 5.

In Fig. 5, the authentication flow diagram for NGS Portal release 2.0 illustrates the fact that a user is authenticated in two ways. First the portal tries to authenticate a user through its own record. If authenticated, the user profile will then be loaded and user role(s) will be set. This is the default NGS Portal login process. If the authentication fails logging as a portal user, the user will then be directed to the MyProxy Login Module and will login as a Grid NGS user. Once the user is authenticated, the user will be mapped to a portal local account and the user's Grid proxy credential will be saved in a separate database rather than modifying the portal's own database. Then the user's role(s) will be set according to StringBeans' record. Once this is performed, the control is then past to StringBeans. Portlets will be able to retrieve the user's proxy credential automatically from the database.

4.3 Inter-Portlet Communication

A big issue in the current JSR 168 standard is that inter-portlet communication is not supported. As defined in the specification, each portlet application is an independent Web application. It is currently only possible to include all portlets inside one portlet application, as we do in the Grid portlet application, to communicate with each other, for instance, to share the proxy credential in sessions. Some portal frameworks vendors like IBM with WebSphere [17] provide inter-portlet communication feature at the portlet application level. In fact, there is always a need to communicate at the portlet application level, i.e., to communicate between portlet applications.

The Java Messaging Service (JMS) [18] has been tested in the BatchJobMonitor portlet. Once a job is finished, a JMS message is sent to a JMS server. This message can then be consumed by any JMS client program. This can be further developed as a notification function. As shown in Fig. 6, a screenshot of the BatchJobMonitor portlet, from which all jobs submitted by the current user are shown, according to the job status, different messages can be sent to notify the end users.

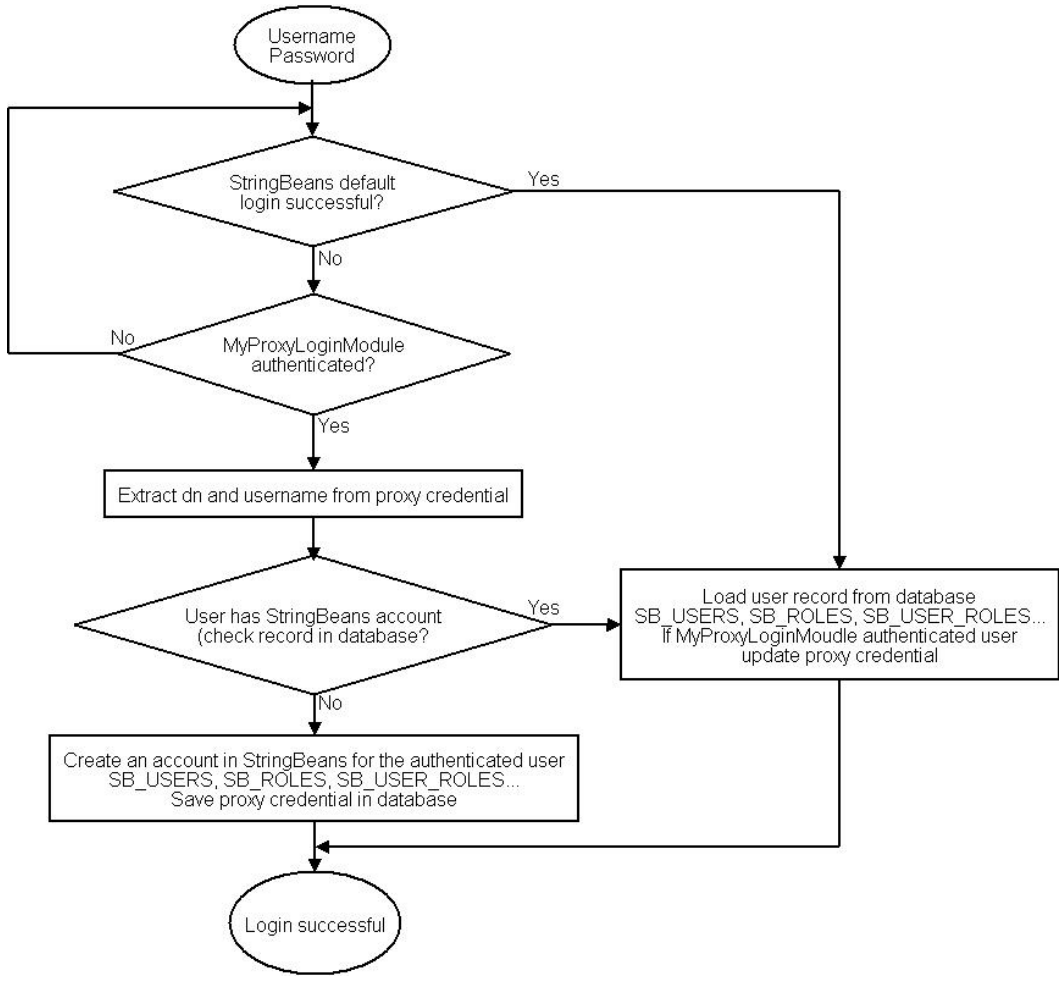


Figure 5: Authentication flow diagram used in the NGS Portal release 2.0

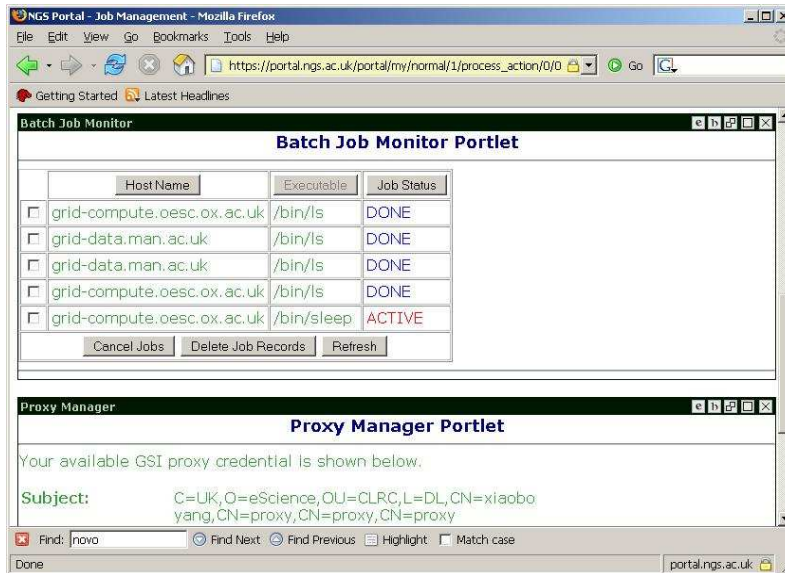


Figure 6: A screenshot of the BatchJobMonitor portlet

4.4 Cloning the Portal for other e-Science Projects

One benefit of adopting the JSR 168 standard is that it is possible to make the Java code reusable since JSR 168 defines a set of standard APIs for portlet development. The Grid portlet application we developed has been successfully deployed in other portals like the IBPortal without modification of the source code itself. We are also investigating the use of WSRP, by which a project may access portlets in the NGS portal remotely.

Projects with which we are currently working to share code are: Sakai VRE Demonstrator, Integrative Biology VRE Demonstrator, e-HTPX e-Science projects, SCARF (a cluster on the CCLRC Facilities Grid) and ReDRESS [19].

5. Conclusions and Future Work

Two releases of the NGS Portal, release 1.0 based on CHEF and release 2.0 based on StringBeans have been described in this paper. A set of Grid related tools/ portlets for UK e-Science users to access the National Grid Service through a user-friendly Web interface has been developed in the Grid Technology Group at CCLRC Daresbury Laboratory. By adopting the JSR 168 standard in release 2.0, the Grid portlet application has been successfully ported to other portal frameworks rather than StringBeans. It has also been deployed for other portals with only modifications in the portlet application configuration files. Whilst SSO through MyProxy has been successfully tested in the NGS Portal release 1.0, by using the MyProxy Login Module, the NGS Portal release 2.0 based on StringBeans 2.4 provides a more standard and flexible way to simplify the usage of the traditional two-step login method to retrieve proxy credentials for end users. The module also illustrates the benefit of adopting JAAS in portal frameworks, which simplifies customisation of portal authentication. With new authentication technologies emerging, it is always possible to adopt them by developing new JAAS login modules, which makes the portal framework automatically support new authentication technologies.

In the future, WSRP will be further investigated to enable the NGS Portal to publish its generic portlets to remote application-specific project portals, which may be using different portal frameworks. This will greatly ease the construction of complex portal based

systems, e.g., Virtual Research Environments (VREs), enabling them to be more scaleable, manageable and maintainable.

Besides JMS which is tested in the NGS Portal release 2.0, the J2EE architecture integrates a lot of powerful components like Enterprise Java Beans (EJBs) and Web Service support, for developing, deploying and managing multi-tier server-centric enterprise applications. The benefits of using J2EE techniques are currently under investigation.

References

- [1] The UK National Grid Service, <http://www.ngs.ac.uk/>.
- [2] S.T. Peltier, A.W. Lin, D. Lee, S. Mock, S. Lamont and M.H. Ellisman, The Telescience Portal for Advanced Tomography Applications, *Journal of Parallel and Distributed Computing*, Vol.63, No.5, 2003, pp. 539-550.
- [3] HotPage Grid Computing Portal, <https://gridport.npaci.edu/>.
- [4] X. Yang, M. Hayes, K. Jenkins and S. Cant, The Cambridge CFD Grid Portal for Large-Scale Distributed CFD Applications, *International Conference on Computational Science 2004 (ICCS2004)*, eds. M. Bubak *et al.*, Springer-Verlag, LNCS 3036, 2004, pp. 478-481.
- [5] A.J. Richards, R.J. Allan, G. Drinkwater and K. Kleese, Building the e-Science Grid in the UK: Advanced Grid Computing, *UK e-Science AHM2003*, Nottingham, UK, 2003.
- [6] R.J. Allan, D. Baker, D. Boyd, D. Chohan, S. Cox, H. Eres, R. Fowler, N. Furmento, J. Giddy, T. Harmer, M. Hayes, N. Hill, J. Hillier, J. Jensen, A. Keane, M. Krznaric, W. Lee, M. McKeown, A. Mills, S. Newhouse, S. Pickles, R. Pining, A. Richards, A. Saleem and J. Watt, Building the e-Science Grid in the UK: Middleware, Applications and Tools Deployed at Levae 2, *UK e-Science AHM2003*, Nottingham, UK, 2003.
- [7] R.J. Allan, C. Awre, M. Baker and A. Fish, *Portals and Portlet 2003 NeSC Workshop*, 14-17 July 2003, http://www.nesc.ac.uk/technical_papers/UkeS-2004-06.pdf.
- [8] Web Services for Remote Portlets (WSRP), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp.
- [9] JSR 168: Portlet Specification, <http://www.jcp.org/en/jsr/detail?id=168>.
- [10] The Sakai Project, <https://www.sakaiproject.org/>.

- [11] Open Grid Computing Environment, <http://www.collab-ogce.org/nmi/index.jsp>.
- [12] Globus Java Commodity Grid Kits, <http://www.globus.org/cog/>.
- [13] J. Novotny, S. Tuecke and V. Welch, An Online Credential Repository for the Grid: MyProxy, *Proceedings of the 10th International Symposium on High Performance Distributed Computing (HPDC-10)*, IEEE Press, pp. 104-111, 2001.
- [14] A. Akram, D. Chohan, X.D. Wang, X. Yang and R. Allan, A Service Oriented Architecture for Portals Using Portlets, *UK e-Science AHM2005*, Nottingham, UK, 2005, accepted.
- [15] StringBeans Portal, <http://www.nabh.com/projects/sbportal>.
- [16] Java Authentication and Authorisation Service, <http://java.sun.com/products/jaas/>
- [17] WebSphere, <http://www-306.ibm.com/software/websphere/>.
- [18] Java Messaging Service (JMS), <http://java.sun.com/products/jms/>.
- [19] R. Crouchley, A. Fish, R.J. Allan and D. Chohan, Virtual Research in the UK: Portal Services for Awareness and Training in e-Science., *UK e-Science AHM2004*, Nottingham, UK, 2004.