

THE CCLRC DATA PORTAL

Glen Drinkwater, Kertsin Kleese, Shoaib Sufi, Lisa Blanshard, Ananta Manandhar, Rik Tyer, Kevin O'Neill,
Michael Doherty, Mark Williams, Andrew Woolf

CCLRC – Daresbury Laboratory, Daresbury, Warrington, Cheshire, WA4 4AD, UK.

E-mail: k.kleese@dl.ac.uk

Abstract: *The project aims to provide easy, transparent access to experimental, observational, simulation and visualisation data kept on a multitude of systems and sites. Further more it will provide links to other web/grid services, which will allow the scientists to further use the selected data, e.g. via data mining, simulations or visualisation. The Data Portal will aim to work as a broker between the scientists, the facilities, the data and other services. The problem addressed is that currently the scientific data is stored distributed across a multitude of sites and systems. Scientists have only very limited support in accessing, managing and transferring their data or indeed in identifying new data resources. In a true Grid environment it is essential to ease many of these processes and the aim of the Data Portal is to help with automating many of these tasks.*

The Data Portal originally used Sun's Java 2 Enterprise Edition (J2EE) but was replaced using a component based web service model.

Key words: Data Access, Globus, SOAP, Web Services, Grid, MetaData.

1. Introduction

Currently the scientists are forced to manually relate between all the experimental, data, computing and analysis facilities that are available world wide, with little infrastructure support. In the future it is hoped the Grid will provide these functions, enabling the scientists to choose much more easily from a wide range of services, connecting and combining desired services for an optimal working environment. Much of the access to the Grid is envisaged to take place through customisable, community oriented Portals. A range of projects within Council for the Central Laboratory of the Research Councils' (CCLRC) have been

chosen to provide the initial building blocks of an integrated solution for users of experimental, computing and data facilities, demonstrating on a few selected examples how basic technologies can be used to build middleware components that support high level scientific grid applications. Data will play a pivotal role in the success of Grid or e-Science developments. Virtually all envisaged applications will need to be able to draw from and deliver to the distributed heterogeneous information/data sources with a variety of contents. Hence three major challenges are posed: data accessibility, data transfer and management of personal data. Data accessibility implies the capability to locate information/data without prior knowledge of its physical location or the form in which its contents is described. Furthermore scientists, as well as applications, need to be able to combine results from different sources. Data transfer relates to the problem of large data volumes that need to be transferred across the Internet. Management of personal data is concerned with the growing distribution of data produced by scientists within a Grid environment, which required new ways of keeping track and moving data for single scientists and more importantly for research groups. CCLRC's integrated data system includes the following components a Data Portal for high-level access to multidisciplinary data, linking to existing data catalogue systems. These catalogues include metadata as well as links to the data itself. The data itself is held in various storage resources from local disks, over databases to multi terabyte tertiary tape systems.

1.1. Current Status

The Data Portal is currently giving access to data from two of our experimental departments: Synchrotron Radiation (SR at Daresbury) and Neutron Spallation (ISIS at RAL) as well as data from the British Atmospheric Data Centre (BADC at RAL). Last year we have also integrated an outside source for demonstration purposes: the Max Planck Institute for Meteorology in Hamburg, Germany. An installation is available at <http://esc.dl.ac.uk:9000/dataportal>.

The Data Portal was designed to work in distributed and heterogeneous environments, our current test installation has proven that to be true, integrating a multitude of systems types, operating systems, databases and sites seamlessly. Furthermore Version 3 of the Data Portal has proven to be robust.

1.2. Architecture

Central to the Data Portal is the Server, which provides the User interface and manages the interaction with the User and all attached resources. The server provides the user with a web interface to search the existing metadata both on the server itself and the connected data holdings transparently. Incoming requests from the user will be interpreted by the server and a query will be formed and transmitted either to the Central Metadata Catalogue or the connected local Metadata Repositories. Currently queries to local repositories have a SQL like notation and are transmitted via http. The result from the various local repositories and the Central Catalogue is expected in XML format. The response generator will collate the results and user output generator will produce the required pages to display the results. The Server is also responsible for the user authentication and session control. In the future the server is also expected to liaise with other data portals as well as other grid services.

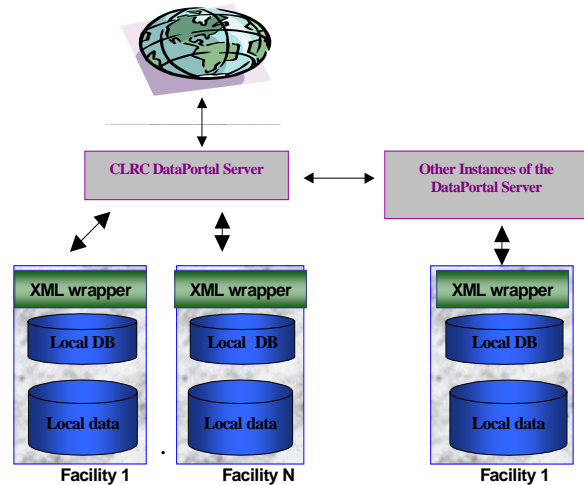


Figure 1. Data Portal General Architecture.

CCLRC has developed a special multidisciplinary metadata format [1] to be able to integrate and make available data from various scientific topics ranging from astronomy to physics. In the following paragraph we will describe how these resources are connected with the server.

Other repositories are expected to have either their own metadata catalogue systems and their own metadata formats describing their data holding or use an extension of the CCLRC Scientific Metadata format (CSMD). To integrate them each catalogue will be accompanied by an XML wrapper, which interprets the requests of the Data Portal server and translates them into local SQL and metadata formats. If the reply consists solely of metadata information the XML output generator and the Response generator will provide a collated XML document to the Data Portal server, with the included metadata translated back into the general metadata format. Does the request also include the wish for actual data, these will be transferred separately and the XML document returned to the server will include the necessary information. Currently the metadata catalogues include links to the data location, the data can be transferred via

GridFTP to the user or to a third party machine.

The core Data Portal system offers the user the possibility to collect all relevant datasets in his personal shopping basket, which can be kept from one session to the next if required. This shopping basket then offers the user a range of functionalities like transfer (using GridFTP), delete (from shopping basket), or if available manipulate. Datasets have to be marked in their metadata database for the Portal to know a manipulation service is available. In the case of the BADC this manipulation is the link to RasDaMan. The user is guided to a manipulation interface, which interacts both with the metadata database (via the XML wrapper) for general information about the dataset as well as the GribServlet, which enables the interaction with the RasDaMan database.

1.3. Version 3

The current version of the Data Portal moves away from the previous implementation which relied solely on Sun's Java 2 Enterprise Edition (J2EE) component based model to a modular web services model. This is achieved using Apache's Axis implementation of the SOAP [2] (Simple Object Access Protocol) submission to W3C. SOAP is a lightweight protocol for exchange of information in a decentralised, distributed environment. It is a XML based protocol, which defines a framework for representing remote procedure calls and responses.

Using SOAP and web services the Data Portal was decentralised into modules that represent an area of functionality. These services were platform and language independent allowing other services (other portals or clients) to communicate with the Data Portal regardless of the language that they were written. Each of the Data Portal modules communicates with each other using SOAP protocol to achieve the same functionality as the previous version.

Vital to this version of the Data Portal is the Lookup module. This is used for the publishing and finding Data Portal web service modules. Essentially this acts as an interface to a Universal Description, Discovery and Integration (UDDI) registry [3]. A module would query the UDDI and receive a Web Services Definition Language (WSDL) file address for the module. This is standard to describe the technical invocation syntax of a web service. A module would use this file to invoke the web service that it needs.

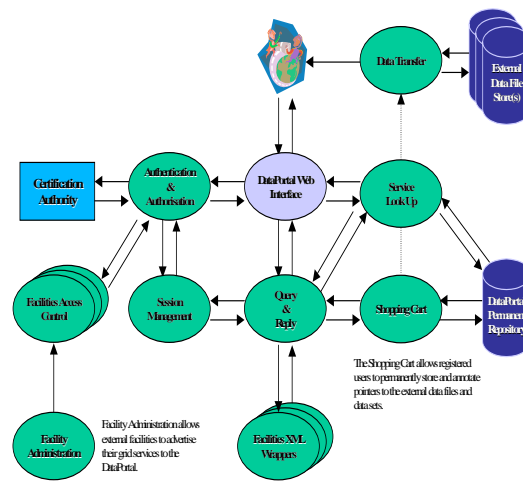


Figure 2. Architectural view.

2. Walk through

The following will give you a simple walk through of some of the key features of the Data Portal.

2.1. Logging On

A user via the web interface would need to authenticate to the Data Portal before searching for data. This is achieved by using an online credentials repository called MyProxy Server [4], requiring a user to have a certificate and a private key associated with the certificate. A user delegates a proxy credential using MyProxy so that they can authenticate themselves to the Data Portal. A

username, proxy passphrase and a credential lifetime are given to the Web Interface and the information passed via SOAP to the Authentication module. The module retrieves another delegated certificate (a self delegated certificate based on the first delegated certificate) out of the MyProxy Server which, in effect, is a short term binding of the user's Distinguished name (DN) to an alternate private key.

Once the user has authorised to the Data Portal, the Authentication module looks up the WSDL addresses for the XML wrappers that are published to the Data Portal. Each wrapper has an associated Access and Control module (ACM), which provides access permissions to the wrapper's data.

Initial log on through Browser / Outside Service

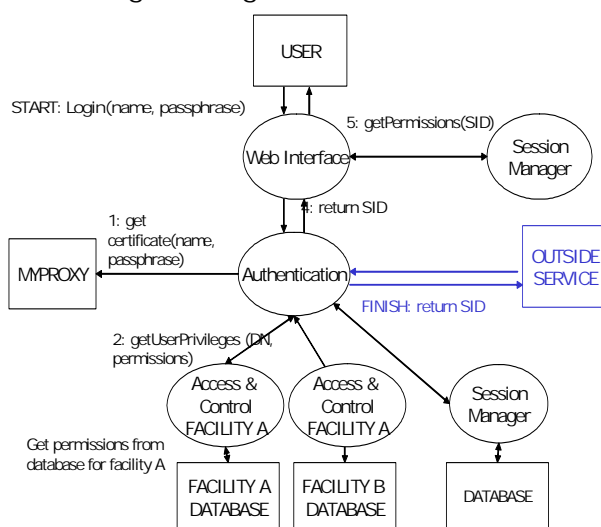


Figure 3. Logging on Process.

The authentication module passes the delegated information to the ACM which uses the DN to decide whether the user has either read access to the XML wrapper's location, metadata or data. Mapping the DN to a local username and obtaining the access rights from the local computer's authorisation mechanism does this. A simple XML document representing the results is sent back to the Authentication module, which concatenates the information from each ACM

and forwards this to the Session Manager module. The Session Manager inserts the information into a database along with the proxy certificate, creates a session id and returns this to the Web Interface via the Authentication module. This completes the logging on sequence.

2.2. Basic Search

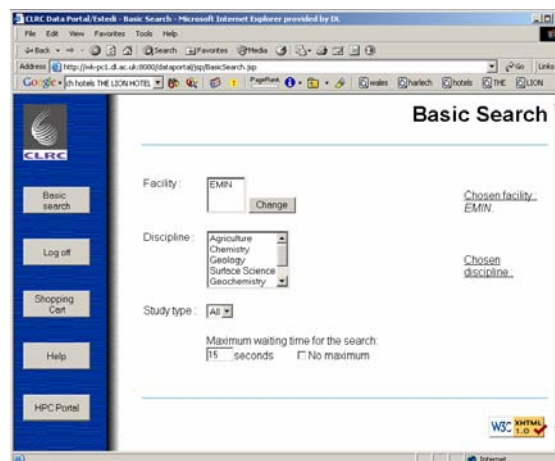


Figure 4. Basic Search Page.

Once the user has successfully logged on to the Data Portal they are ready to search for data. The Web Interface queries the Session Manager for the access rights of the user and displays a list of the XML wrappers that the user has read access to. The user drills down to select a topic and sends the query to be processed by the Query and Reply module. If the session id is still valid (i.e. the proxy certificate has not expired) the module queries the Lookup module to locate the WSDL files for the XML wrappers that the user has chosen. The XML wrappers translate the query into a local querying language such as Xpath or SQL. The local database(s) are searched and return the results. This is transformed into CCLRC's scientific metadata format in XML. Each wrapper returns the XML document to the Query and Reply, which is concatenated into one XML file and returns this to the Web Interface.

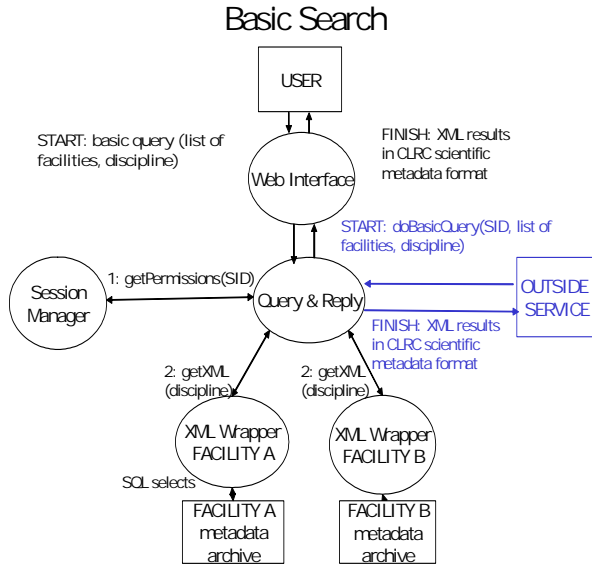


Figure 5. Basic Search - Modules used.

Initial search results from the basic or advanced search will be displayed to the user. The ones of most interest they can expand further. Finally they will be able to select useful datasets/files for further processing.

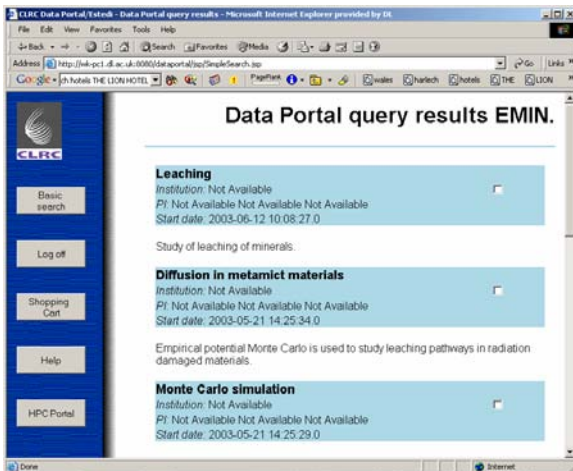


Figure 6. Initial Result Page.

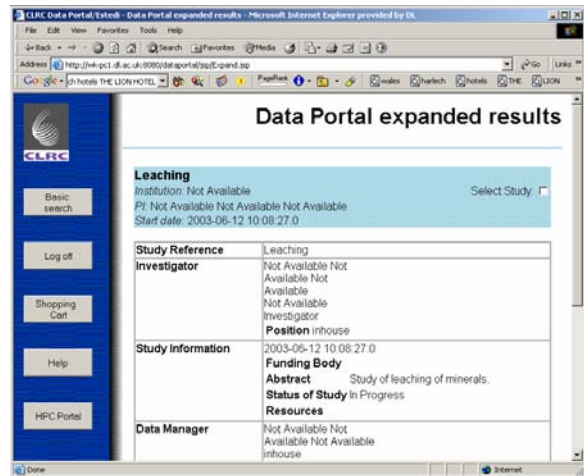


Figure 7. Expanded Result Page.

2.2. Shopping Cart

The XML results are displayed to the user via an Extensible Stylesheet Language Transformation (XSLT). The user narrows down their search until they wish to add studies, datasets or individual files to the shopping cart. This represents a persistent data storage of the user's choices over previous sessions from the Data Portal. The Web Interface collects the information from the CCLRC XML metadata format from the choices that the user has given, sending a XML representation of the items added to the Shopping Cart module.

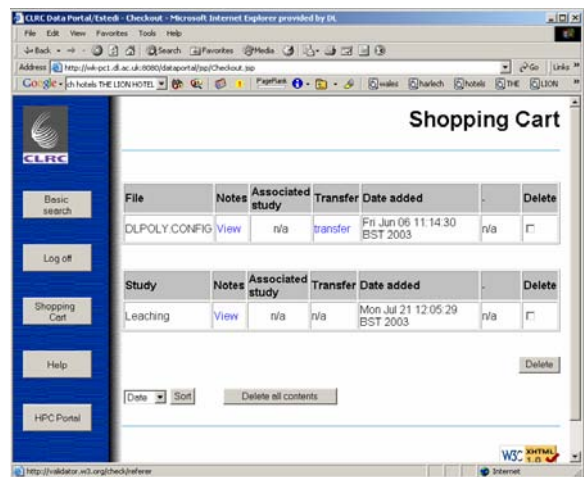


Figure 8. The Shopping Cart.

The Shopping Cart passes the session id to the Session Manager, which returns the user's DN. The DN is mapped to the persistent shopping cart for the user and adds the items. The user is then directed to the page that displays the contents of their shopping cart. The Shopping Cart can offer a range of functionalities e.g. third party transfer of files or datasets and in the case of the BADC, a manipulation link to RasDaMan [5], which connects to the RasDaMan database, manipulates the data and returns the results.

The user can also decide to transfer a single file or a move a complete dataset to another location. The shopping cart stores the location(s) of the file or the dataset's files for the user to transfer the data whenever they wish. The Web Interface would pass the file location(s) information to the Data Transfer module which would request the user's certificate from the Session Manager via the session id. The Data Transfer module would transfer the files using the user's proxy certificate as the credentials for the transfer, allowing the user to move the data to any machine with the user's DN in the globus grid map file [7]. The gridmap file basically maps a DN to a local user on the machine.

2.3. Outside Services

One of the key reasons for moving to web services is for the ability for other applications and web services to access the Data Portals services. This is independent of the programming language that the web services are written in. At CCLRC two other portals are been developed, the HPC portal and Visualisation portal. One possible scenario is that a user has found some data using the Data Portal and wishes to run an application on the data and visualise the results. The web service infrastructure of the Data Portal allows other applications to communicate with the modules. The HPC portal could communicate with the shopping basket on behalf of the user. Once the location of the data was extracted from of the

cart, the data could be transferred to a machine where the application resides. Once the application finishes the results can then be visualised via the Visualisation portal which can be invoked again using web services.

Single sign on between the HPC Portal and the Data Portal can be achieved via session information being shared between the separate Session Managers of the portals. A user wishing to use the functionalities available to the Data Portal e.g. a file URL in their shopping cart, who is coming through the HPC Portal, could use their proxy certificate stored in the HPC Session Manager to authenticate them to the Data Portal. This is achieved via mutual authentication between the two web servers. Once the Data Portal Session Manager has authenticated that the client is the HPC Portal it trusts that the proxy certificate being passed is a valid proxy and starts a session on the Data Portal. The proxy certificate is transferred encrypted and is therefore safe for internet transfer via web services.

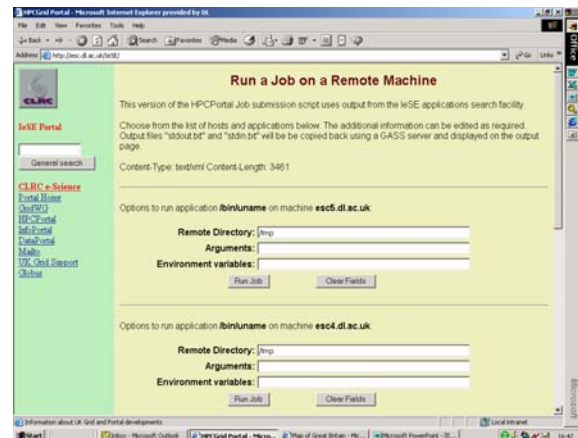


Figure 9. HPCPortal.

With single sign on between the portals a user could access the data in their shopping cart, transfer the data a run a job on the data or vice versa, coming through the Data Portal first, without having to logon twice or have their MyProxy server username and passphrase passed via web services.

In the unlikely case of the proxy certificate been viewed by an outside agent the life time of the proxy certificate is so short (1-2 hours) that it would expire before the agent is able to use it.

3. Current Work

3.1 Security

Currently authentication within the Data Portal is done by the exchange of the user's proxy certificate to the ACM for each facility. The facility maps the user's DN to a local user and the access rights are given back to the Data Portal in the form of an XML document. The XML document gives information regarding the read access to the facility and then read access to the data and metadata respectively.

The new architecture developed is slightly different to the previous. Each XML Wrapper has an associated Access Control Modules. The ACM signs the XML document with its private key which the Data Portal stores in the Session Manager's database. When the Query and Reply modules sends a request to the XML Wrapper it now sends the XML document associated with the facility. The XML Wrapper can validate the signature of the XML document with the ACM's public key. Therefore the XML Wrapper can trust the access information regarding the facility given in the XML document.

3.2 XML Wrapper

The XML Wrappers are used to convert between the local metadata format of a data archive and the CCLRC Scientific Metadata format. The wrappers represent a common interface to the Data Portal core modules (e.g. Query and Reply module) which allows the Data Portal to seamlessly interact with the different data archives. While essentially a wrapper can be written in any language as long as they expose the same

interface it is of interest to talk about the architectures involved in writing them as we are moving towards a new more flexible architecture.

The current architecture takes in a restricted 'SQL' like where clause and returns the CSMD records which match the results of the query - i.e. the whole study metadata which matches the results. The wrapper itself selects the study keys from the Data Archive which match the 'where clause' and either builds the record in real time (which is slow) or retrieves it from the cache if it has already been built.

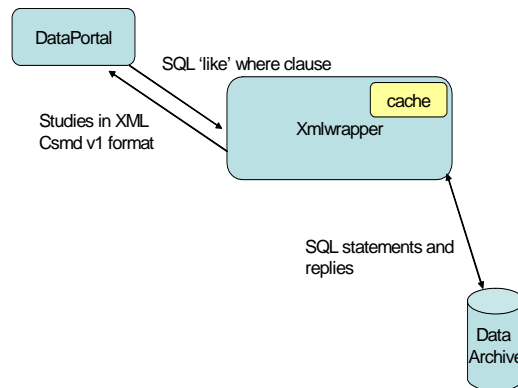


Figure 10. Old Architecture

The new XML Wrapper Architecture uses a different approach. The XML Wrapper interface now takes in an XQUERY and the XML Wrapper Document Selector (XWDS) applies this query to the document repository and returns the results that are specified by the XQUERY.

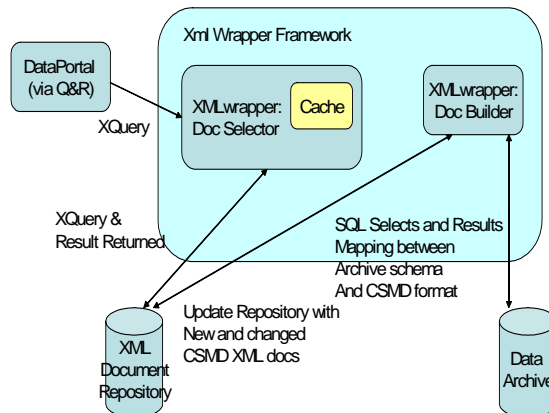


Figure 11. New Architecture

This version 4 architecture has a number of advantages, firstly a bespoke API for querying the data is not needed to be specified or supported, the CSMD XML format is known and XQUERY's are logical to write and will have a form which is subject to standardisation. Also the processing is done by the Wrapper to select only the fragments from the studies that are needed, formatting information can also be specified in the XQUERY such that results returned now contain only the information needed for the level of detail of presentation and are in the correct format needing minimal amendment for presentation. This approach has two advantages, the transfer of huge studies and then the huge processing overhead of applying XSLT stylesheet to all these studies by the Data Portal is removed. Also, in the queries processed by the XWDS key selection is independent of the data archive and hence information can still be retrieved if the archive is unavailable on the network. The work done by the XWDS is also separate from that done by the Data Portal allowing more distributed processing, a more Grid based approach. The architecture also allows for a separation of concerns the issue of building the CSMD files and cache coherency are now in a separate module the XML Wrapper Document builder (XWDB). The flexibility offered by this new approach is hoped to allow a more efficient and scalable effect on the Data Portal as a whole.

4 Future

The Data Portal is will be taking account of new technologies (e.g. OGSA [7] and the Globus Toolkit 3 (GT3) [7][8]). Further work and research will be undertaken with other projects and the new technologies mentioned above. This will allow other modules will be added to this release e.g. notification of web service completion for long queries or transfers.

The Data Portal technology will also be used for a range of projects e.g.

- Environment from the Molecular Level (NERC)
- Simulation of new Materials (EPSRC)
- NERC DataGrid (NERC)

In the NERC DataGrid project we will aim to integrate our existing software with the packages of the US Earth Systems Grid. First discussions have shown the developments to be complementary and future perspectives to be very similar.

Heavy emphasise will be put in the future on the developments of metadata for various aspects of the portal and ontologies, which we think are essential to sustainable progress.

5 References

- [1] Matthews BM, Sufi SA. The CCLRC Scientific Metadata Model - Version 1. The CCLRC Scientific Metadata Model - Version 1 2002.
- [2] Box D, Ehnebuske D, Kakivaya G, Layman A, Mendelsohn N, Nielsen HF, Thatte S, Winer D. Simple object access protocol (SOAP) 1.1. 2000 <http://www.w3.org/TR/SOAP/>
- [3] Allan RJ, Wang SD, Chohan D, McKeown M, Colgrave J, Dovey M. UDDI and WSIL for e-Science. UK Grid Support Centre 2002 <http://esc.dl.ac.uk/WebServices>
- [4] Novotny J, Tuecke S, Welch V. An online Credential Repository for the Grid: MyProxy <http://www.forwiss.tu-muenchen.de/~rasdaman/>
- [5] K.Kleese, P.Baumann: Intelligent Support for High I/O Requirements of Leading Edge Scientific Codes on High-End Computing Systems - The ESTEDI Project, Proceedings of the Sixth SGI/Cray MPP Workshop, p.167-178 B. J. Jesson (Eds), Manchester University, Manchester, UK, 2000
- [7] The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. I. Foster, C. Kesselman, J. Nick, S. Tuecke, Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002. <http://www.globus.org/research/papers/ogsa.pdf>
- [8] A Java Commodity Grid Toolkit. G. von Laszewski, I. Foster, J. Gawor, P. Lane. *Concurrency: Practice and Experience*, 13, 2001. ftp://info.mcs.anl.gov/pub/tech_reports/reports/P823.ps.Z