Computational Grid Portals in Virtual Engineering Design Environment

A.B.M. Russel
Monash University
E-mail: abm.russel@its.monash.edu.au

Thomas Ting, Chris Seeling, Sudarshan Ramachandran
Victorian Partnership for Advanced Computing
E-mail: {sram, thomas, chs}@vpac.org

Abstract

Computational grid portals are becoming essential tools in automotive companies world-wide for managing compute resources for virtual engineering and design. Many automotive and associated manufacturing industries are taking initiatives for making the computing resources transparent from the integrated design environment. This paper illustrates how the grid portal technologies can fulfil these enterprise development needs. We demonstrate the power of Grid portal and Grid middleware technology by developing a computational Grid portal and integrating it with Computer Aided Engineering (CAE) applications in the virtual engineering environment. Using the developed web-based portal, engineers can use the Grid recourses anytime from anywhere. A framework/toolkit is also developed which is kept in front of the enterprise firewall to make any kind of access to the computational Grid resources completely secure from the user. By integrating this portal toolkit with the CAE applications, engineers are able to use the Grid resources completely transparently which lets them concentrate more on their respective jobs from remote location.

1. Introduction

Computational grid portals are becoming vital tools in automotive companies world-wide to simplify access to the compute resources for virtual engineering and design. Many automotive and associated manufacturing industries are taking initiatives for making the computing resources transparent from their integrated design environments. Transparency of the Grid resources will benefit engineers to concentrate more on their respective jobs from remote location. In these working environments of engineers the Grid portals can have a significance role for integration of the emerging web/grid services into the Computer Aided Engineering (CAE) applications. But developing these portals, integrating and maintaining distributed Grid resources altogether requires a significant amount of effort [1, 7, 26, 27]. Thus the usage of computational Grid portal in engineering design environment will not only shorten the development time & cost significantly but also will reduce extra burden of maintaining local/remote compute resources by the engineers.

A web based Grid Portal [2, 17, 19, 24, 25] allows engineers to access Grid resources from a conventional web browser. In Europe many manufacturing industries have already taken similar initiatives for accessing compute resources using any web browser. But applications and implementation of similar technologies in the Australian automotive and associated manufacturing industries have been slow or non-existent. The reason for this slowness is for handling the existing complexities produced in the virtual integrated design environment and at the same time managing high performance computing resources from the design environment is complex & costly. Also developing a suitable framework for integrating different technologies with various APIs and coordinating distributed compute resources requires extra efforts. There is a need to develop a framework to ease the tasks of engineers while working with numerous Computer Aided Engineering (CAE) tools. Thus the Centre for Computational Prototyping of Victorian Partnership for Advanced Computing (VPAC) Ltd. took the first initiative to develop a framework and toolkit for Australian automotive or related industries for accessing the computational Grid resources transparently from the design environment.

The rest of this paper first discusses related work in Section 2, and then describes our solution in Section 3.

Section 4 describes some status of the framework. Section 5 presents our conclusions and describes our future work.

2. Related Work

Virtual Engineering Design is becoming popular in manufacturing industries for reduced time to market, reduced cost and improved quality. The aim of virtual engineering is to integrate a collection of engineering tasks into a virtual environment and assisting clients in maintaining innovation and remaining at the forefront of engineering advancements. It provides a collaborative framework [21] to integrate all of the models, data, and tools needed for good engineering design to Grid systems [9]. To display and analyse engineering data in virtual environments, many companies use CAE tools such as LS-Ddyna, Nastran, StarCD, Fluent etc. together with other CAE systems in preference to physical testing for rapid productivity and profitability.

Foster and Kesselman defined a computational grid [13] as a hardware and software infrastructure that provides access to data, computational, and human resources distributed over wide area environments. This environment is a collection of Virtual Organization (VO) [14] which is a set of individuals or institutions or organizations. Computational Grid is a standard infrastructure for high performance computing. However, the original idea behind grid computing was to support collaborative problem solving in virtual organizations (VO).

At present the Globus Toolkit [16] is the most popular tool for developing standards for the grid infrastructure coordinated world-wide by the Global Grid Forum. The latest version of the Globus Toolkit (GT4) [16] is an extension of the industry standard Web Services specification, known as Web Services Resource Framework (WSRF) [15] jointly proposed by IBM, ANL and GGF. To fulfil the enterprise needs the Grid infrastructure is presently integrated with web services standards [12].

A portal is a web server that acts a front entrance to the Grid systems. A Grid portal is a user's point of access to the Grid systems. It provides an environment where the user can access Grid resources and services, execute and monitor Grid applications, and collaborate with other users. Other key requirements are:

- **Security services:** Once the user is authenticated to the portal server, the portal server accesses the grid services on behalf of the user [8].
- **Remote file management services:** User is able to browse the remote data archives using the portal and manipulate the data from the remote location.
- **Remote job management services:** User is able to submit & monitor the jobs to the Grid. User is able to see the job queues too.
- **Application interfaces services:** The portal is able to hide Grid details behind useful application interfaces. The user needs to be able to launch, configure and control remote applications in the same way he or she uses desktop applications. The portal must provide a way to easily integrate new application interfaces and expose them to the user and hide the details of Grid middleware.

Examples of portals and similar tools are LEAD for predicting severe storms, NEES Grid for earthquake engineering, BIRN for medical research, GEON for geosciences research, TeraGrid user portal for providing access to the US TeraGrid, LEGION portal [22], GENIUS Portal [6] Nimrod/G portal [23], Enginframe Grid middleware of NICE etc.

3. Our Solution

To reduce the development time & cost we decided to develop the framework/toolkit based on the Enginframe grid middleware [10]. The Enginframe middleware is built on XML technology that supports next generation browsers and web access devices. It provides secure access using SSL and VPN tunneling and integrated with ICA protocol and X-protocol for LAN/WAN access to Windows and UNIX applications. It is built on the top of existing web servers [3, 4] and supports of Grid infrastructure with data transfer facilities including remote file browsing.

3.1. Portal Architecture

Along with the Enginframe middleware, the overall architecture of the VPAC Engineering Grid portal framework is logically structured into three different tiers:

- **Client Tier:**

This tier consists on the user's browser and the vertical application software. This tier supports next generation browsers along with web access device (e.g. WML/GPRS/UMTS cell phones, wireless PDA, etc.). This tier is responsible for authentication, data management, job submission and job tracking etc.

- **Server Tier:**

This middle tier consists of servlet-enabled web server [3, 4] and Enginframe middleware server. This tier provides the resource brokering and management activities. Also this tier forwards authentication credentials, manages integration of data and services from different services and resources.

- **Resource/Agent Tier:**

This tier is the resource tier consists of Agents (clusters, standalone hosts) to control the computing resources. This tier provides distributed access to existing computing resources and uses Grid infrastructure for execution of any service request from client.

We propose the following infrastructure for computational resources management architecture for engineering jobs on grid resources:

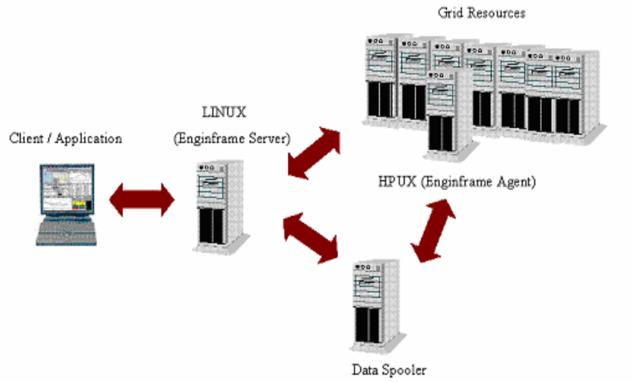


Figure 1: High level architectural view of VPAC Computational Grid Portal

3.2. Portal Services

Initially we identified the required common services used by CAE applications to access high performance compute resources. Some of the services of are, authentication service, Grid Load Viewing Service, Job Listing Service, Data Viewing Service, Job Submission Service, Data Transfer Service, Reporting Service, Project Management Service etc. Among all these services authentication service handles the user's authentication. It provides session management for each authentication through HTTP connection using web-browser and flow management for each authentication. It uses SOAP (Simple Object Access Protocol) [18] as the message transport protocol. SOAP uses Extensible Markup Language (XML) [11] to perform remote procedure calls for any transaction among web services. XML is used to create the messages sent both to the web service from the client, and again for the message response sent from the web service back to the client. Other services like Grid load viewing service provides user overall information about the current load at the entire Grid system. This will let the user know about the percentage of the current load and the number of free CPUs available and all jobs running at the Grid system. Job listing service provides the list of user's all running/queued jobs at the Grid system. Data transfer service provides directory creation, content/data/file uploading facilities into the spooler directory. Figure 2 shows data viewing service provides the list of user's all data (outputs from the jobs, uploaded files etc.) from the current spooler. Job submission service facilitates the submission of jobs (LS-Dyna, Nastran, Fluent, StarCD etc.) on the Grid system transparently.

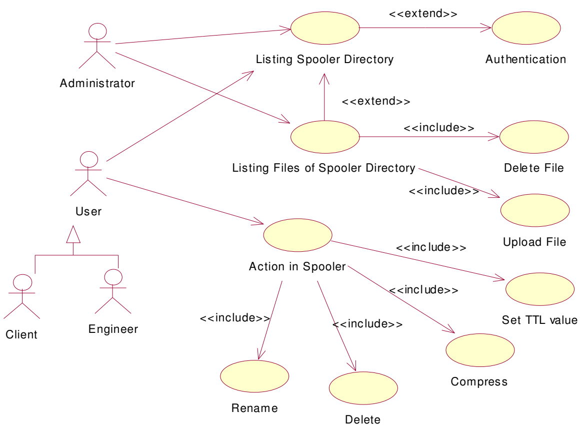


Figure 2: Data viewing service example for design purpose

3.3. Portal Development

With the help of several complex distributed technologies and Enginframe Grid middleware we successfully developed the VPAC Engineering Grid portal. Initially, all web services are created using XML/XSL framework at the web server tier along with the Enginframe server. At the Agent server, services are implemented with XML and shell/AWK scripts. Each scripts responses in a GridML (XML) format according to the requests from the server. At the server end these GridML are processed and with the help of particular XSL the output is rendered into proper XML/HTML format and delivered to the browser.

4. Results

With the power of Enginframe Grid middleware and related distributed technologies we successfully developed the VPAC Engineering Grid portal framework. While developing this portal we also able to solve the following enterprise requirements in the virtual engineering design environment.

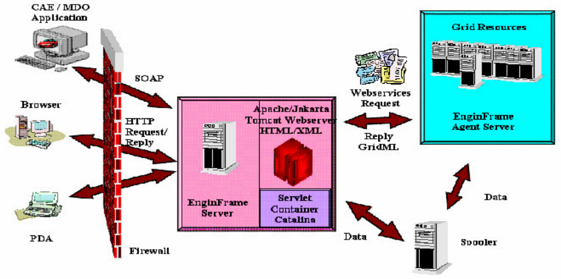
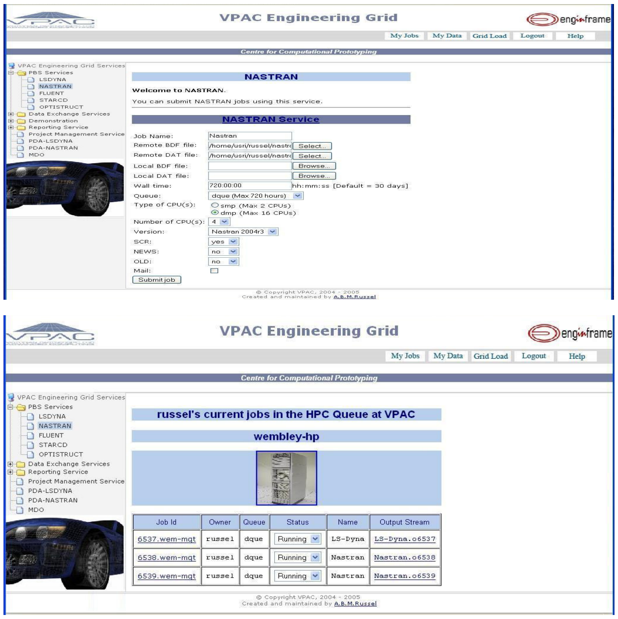


Figure 3: VPAC Computational Grid Portal architecture

4.1. Job Submission

The job submission service submits LS-Dyna, Nastran, Fluent, StarCD, Optistruct etc. job submission on the Grid resources. This service requires authentication first. User need to provide necessary information and need to press submit button to submit a successful job on the Grid.



NASTRAN

Welcome to NASTRAN.

You can submit NASTRAN jobs using this service.

NASTRAN Service

Job Name:	Nastran
Remote BDF file:	/home/usri/russel/nastr
Remote DAT file:	/home/usri/russel/nastr
Local BDF file:	
Local DAT file:	
Wall time:	720:00:00 hh:mm:ss [Default = 30 days]
Queue:	dque (Max 720 hours)
Type of CPU(s):	○ smp (Max 2 CPUs) ● dmp (Max 16 CPUs)
Number of CPU(s):	4
Version:	Nastran 2004r3
SCR:	yes
NEWS:	no
OLD:	no
Mail:	□

© Copyright VPAC, 2004 - 2005
Created and maintained by A.B.M.Russel

russel's current jobs in the HPC Queue at VPAC

wembley-hp

Job Id	Owner	Queue	Status	Name	Output Stream
6537.wem-mgt	russel	dque	Running	LS-Dyna	LS-Dyna.o6537
6538.wem-mgt	russel	dque	Running	Nastran	Nastran.o6538
6539.wem-mgt	russel	dque	Running	Nastran	Nastran.o6539

© Copyright VPAC, 2004 - 2005
Created and maintained by A.B.M.Russel

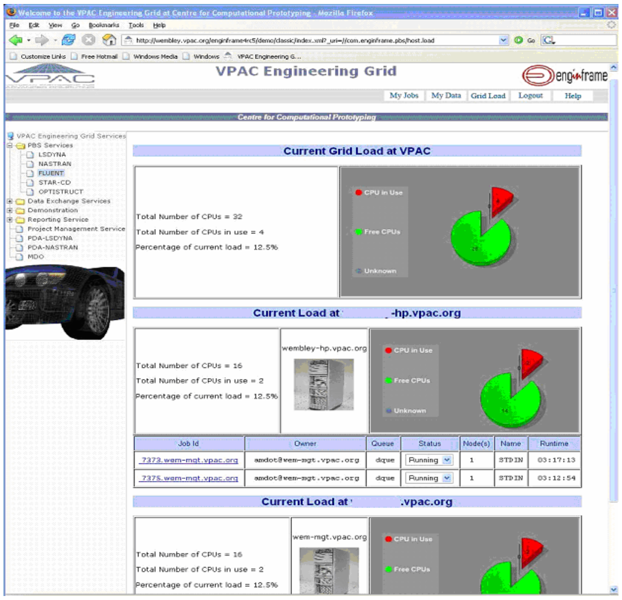
Figure 4: Job submission service and jobs listing service portal

4.2 Dynamic job submission script generation

This was one of the key issues in this project. Instead of having a common script for executing Grid service requests we propose a dynamic script generation policy. This allows us to reuse the similar services using the same script too. We also categorise the scripts by the type of CPUs. For example, we use qsub-JOBTYPE-CPUTYPE script where JOBTYPE could be LS-Dyna, Nastran, StarCD, Fluent etc and CPUTYPE could be smp, mpp, dmp etc. Similarly for other services we use separate scripts. This script then generates required PBS scripts job submission when any request comes from the client.

4.3 Viewing Grid load.

Another challenge in this project was to represent the Grid load graphically from the user's perspective. To resolve this we have created a new service for viewing Grid load. This service calls required scripts at the Agent and gets results in GridML format which then rendered using XSL into the desired format and displayed on the browser. To do this rendering a Chart.xsl file is created which uses a pre-defined xml file with swf plug-in to render the GridML data into Flush output. The Grid load viewing service displays the information of the current Grid load on the clusters. This service also displays a 3D pie chart with the options: CPU in use, Free CPUs and Unknown. This service also includes cluster information.



Current Grid Load at VPAC

Total Number of CPUs = 32
Total Number of CPUs in use = 4
Percentage of current load = 12.5%

Current Load at -hp.vpac.org

Total Number of CPUs = 16
Total Number of CPUs in use = 2
Percentage of current load = 12.5%

Job Id	Owner	Queue	Status	Node(s)	Name	Runtime
7373.wem-mgt.vpac.org	amdot@wem-mgt.vpac.org	dque	Running	1	STDIN	03:17:13
7375.wem-mgt.vpac.org	amdot@wem-mgt.vpac.org	dque	Running	1	STDIN	03:12:54

Current Load at .vpac.org

Total Number of CPUs = 16
Total Number of CPUs in use = 2
Percentage of current load = 12.5%

Figure 5: Grid load viewing for multiple clusters

4.4 Grid Portal PDA Access.

With the emergence of wireless & mobile technology we develop our portal compatible with PDA device. We have successfully demonstrated the PDA access with the help of Enginframe middleware. Due to the limited energy of mobile/PDA device we also defined services explicitly with very limited information for job submission.

Figure 6: Job submission service using PDA device

4.5 Pre-processing & post-processing facilities

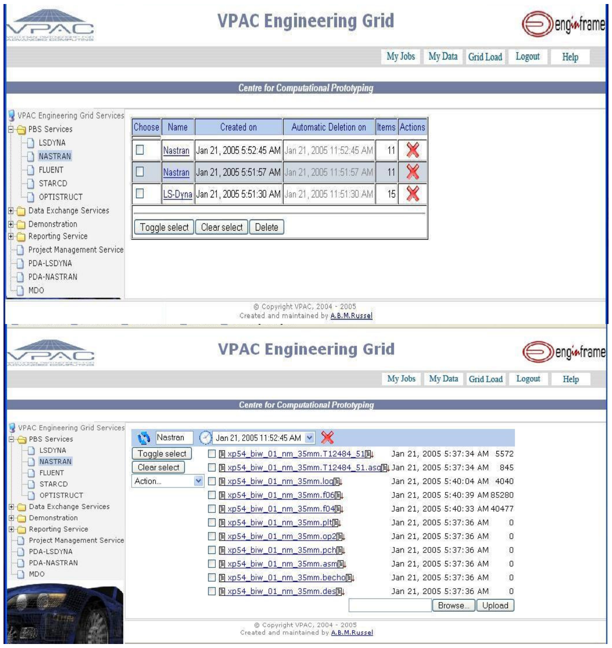
We have successfully implemented the pre-processing & post-processing compatibilities with the Grid portal. We can choose pre-processing or post-processing option while submitting the jobs like Fluent on the Grid.

4.6 Exception handling

We have successfully identified and implemented all exceptional issue might come while submitting a Grid job. Examples are, missing input files; some CPU types can not choose more than 2 CPUs etc.

4.7 Data transfer & spooler data management

With the power of Enginframe middleware, data upload/download files to/from spooler service has been successfully implemented. For the web based portal this service includes actions in spooler data like archiving the files and then download. This archiving will reduce the size of data which are required to download at user's end. For the application based toolkit, we maintained the id of each connection along with every directory created while uploading data at the server end. This allows user to keep track with their jobs outputs and can download the right output files from the server. We have also handled all kinds of issue related to the spooler. This includes, displaying spooler directories, displaying directory contents, browsing files and its contents, creating directory/files in the spooler.



Choose	Name	Created on	Automatic Deletion on	Items	Actions
□	Nastran	Jan 21, 2005 5:52:45 AM	Jan 21, 2005 11:52:45 AM	11	✗
□	Nastran	Jan 21, 2005 5:51:57 AM	Jan 21, 2005 11:51:57 AM	11	✗
□	LS-Dyna	Jan 21, 2005 5:51:30 AM	Jan 21, 2005 11:51:30 AM	15	✗

□	xp54_biw_01_nm_35mm.T12484_51	Jan 21, 2005 5:37:34 AM	5572
□	xp54_biw_01_nm_35mm.T12484_51.asq	Jan 21, 2005 5:37:34 AM	845
□	xp54_biw_01_nm_35mm.log	Jan 21, 2005 5:40:04 AM	4040
□	xp54_biw_01_nm_35mm.f06	Jan 21, 2005 5:40:39 AM	85280
□	xp54_biw_01_nm_35mm.f04	Jan 21, 2005 5:40:33 AM	40477
□	xp54_biw_01_nm_35mm.plt	Jan 21, 2005 5:37:36 AM	0
□	xp54_biw_01_nm_35mm.op2	Jan 21, 2005 5:37:36 AM	0
□	xp54_biw_01_nm_35mm.pch	Jan 21, 2005 5:37:36 AM	0
□	xp54_biw_01_nm_35mm.asm	Jan 21, 2005 5:37:36 AM	0
□	xp54_biw_01_nm_35mm.becho	Jan 21, 2005 5:37:36 AM	0
□	xp54_biw_01_nm_35mm.des	Jan 21, 2005 5:37:36 AM	0

Figure 7: Data management service portals

4.8 Integration with non Java Client with the Grid portal Toolkit

A framework/toolkit is also developed using ANT [5] and Java along with the Java based Enginframe middleware [10] web service library. To make the Java based toolkit compatible with other programming languages (VC++, Python etc.) based client applications, Java Native Interface (JNI) based toolkit JACE [20] is used. We have demonstrated this integration with the Java based Grid portal toolkit with a 3D VirTools engine based VC++ application which was a part of Multi-Disciplinary Optimisation project.

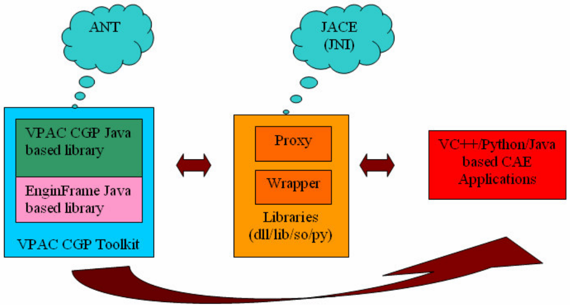

ANT
JACE (JNI)
VPAC CGP Java based library
EnginFrame Java based library
VPAC CGP Toolkit
Proxy
Wrapper
Libraries (dll/lib/so/py)
VC++/Python/Java based CAE Applications

Figure 8: VPAC Computational Grid Portal Toolkit

5. Conclusion and Future work

This paper demonstrates that using the web based Grid portal, an engineer can access the Grid resources from anywhere anytime using PC, laptop or even any mobile or PDA devices. The portal toolkit can also be integrated with any client applications written in Java, VC++, Python etc. We are continuing to investigate further to add more functionality with our VPAC Engineering Grid portal. We expect that with the power of web services and Grid middleware technologies we would be able to extend the computational Grid portal to fulfil the enterprise needs in Virtual Engineering Design Environments. Our future work would be:

- Integrate more clusters with VPAC Engineering Grid portal
- Integrate the toolkit with multiple languages based client applications
- Implementing optimistic load balancing policies among clusters
- Improve data transfer time & efficiency
- Automatic management of remote data.
- Full integration with the Multidisciplinary Optimisation (MDO) project.

6. Acknowledgement

This project was fully supported by Victorian Partnership for Advanced Computing (VPAC). We are thankful to Professor Bill Appelbe, CEO of VPAC for his kind suggestions at various stages of this project (SI0402). We are also grateful to many engineers and researchers of VPAC, Australia and Enginframe developers of NICE, Italy for their vigorous cooperation.

7. References

[1] Akarsu, E., Fox, G., Haupt, T., Kalinichenko, K., Kim, K., Sheethalnath, P., and Youn, C. “Using Gateway System to provide a desktop access to high performance computational resources”. Proceedings of the Eight IEEE International Symposium on High Performance Distributed Computing, August, 1999, IEEE Computer Society, Washington, DC, USA.

[2] Aloisio, G., Cafaro, M., “Web-based access to the Grid using the Grid Resource Broker Portal”, Concurrency and Computation: Practice and Experience Journal, Vol. 14, Grid Computing environments Special Issue 13-14, 2002.

[3] Apache Web Server. Available online: http://www.apache.org/

[4] Apache Jakarta.
Available online: Tomcat http://jakarta.apache.org/tomcat/

[5] Apache ANT.
Available online: http://ant.apache.org/

[6] Barbera, R., Falzone, A., Rodolico, A., “The GENIUS Grid Portal”, Computing in High Energy and Nuclear Physics, 24-28 March 2003, La Jolla, California.

[7] Berman, F., Fox, G., Hey, T., “Grid Computing: Making the Global Infrastructure a Reality”, John Wiley & Sons Ltd, Chichester, 2003.

[8] Butt, A., Adabala, S., Kapadia, N., Figueiredo, R., Fortes, J., Grid-computing portals and security issues, Journal of Parallel and Distributed Computing, Volume 63 , Issue 10, October 2003, Scalable web services and architecture, Pages: 1006 – 1014.

[9] Chen, L., Cox, S., Goble, C., Keane, A., Roberts, A., Shadbolt, N., Smart, P., Tao, F., “Engineering Knowledge for Engineering Grid Applications”, Proceedings of EuroWeb, The Web and the GRID: from e-science to e-business, Oxford, 2002.

[10] Enginframe Middleware.
Available online: http://www.enginframe.com/

[11] Extensible Markup Language (XML).
Available online: http://www.w3c.org/XML

[12] Foster, I., Kesselman, C., Nick,J., Tuecke, S. , The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

[13] Foster, I., Kesselman, C., “The Grid: Blueprint for a New Computing Infrastructure”, First Edition, Morgan Kaufmann Publisher, 1999.

[14] Foster, I., Kesselman, C., Tuecke, S., “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”. International Journal of High Performance Computing Applications, 2001, volume 15, issue 3, pp. 200–222.

[15] Global Grid Forum, Edinburgh, Scotland.
Available online: http://www.gridforum.org/

[16] Globus Toolkit.
Available online: http://www-unix.globus.org/toolkit/

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

[18] Simple Object Access Protocol (SOAP).
Available online: http://www.w3c.org/TR/SOAP

[19] Haupt, T., Bangalore, P., Henley, G., “Mississippi Computational Web Portal”, Concurrency and Computation: Practice and Experience, Vol. 14, Grid Computing environments Special Issue 13-14, 2002.

[20] JACE project.
Available online: http://sourceforge.net/projects/jace/

[21] Mann, V., Parashar, M., "Engineering an Interoperable Computational Collaboratory on the Grid", Concurrency and Computation: Practice and Experience, Vol. 14, Grid Computing environments Special Issue 13-15, pages 1569-1594, 2002.

[22] Natrajan, A., Nguyen-Tuong, A., Humphrey, M., Grimshaw, A, "The Legion Grid Portal", Concurrency and Computation: Practice and Experience Vol. 14, Grid Computing environments Special Issue 13-14, 2002.

[23] Nimrod/G Portal. Available online:
http://www.csse.monash.edu.au/~nimrod/nimrodportal/

[24] Novotny, J., "The Grid Portal Development Kit", Concurrency and Computation: Practice and Experience Vol. 14, Grid Computing environments Special Issue 13-15, pages 1129-11444, 2002.

[25] Pierce, M., Fox, G., Youn, C., Mock, S., Mueller, K., Balsoy, O. "Interoperable web services for computational portals", Proceedings of ACM/IEEE conference of Super Computing 2002.

[26] Thomas, M. P., Mock, S., Dahan, M., Mueller, K., Sutton, D., "The GridPort Toolkit: a System for Building Grid Portals", Proceedings of the Tenth IEEE International Symposium on High Performance Distributed Computing, August, 2001.

[27] Thomas, M., Dahan, M., Mueller, K., Mock, S., Mills, C., Regno, R., "Application Portals: Practice and Experience", Concurrency and Computation: Practice and Experience, Vol. 14, Grid Computing environments Special Issue 13-15, pages 1427-1444, 2002.