Collaborative Multidisciplinary Design in Virtual Environments

Toan Nguyen, Vittorio Selmin

Collaborative Multidisciplinary Design in Virtual Environments

Toan Nguyen[1], Vittorio Selmin[2]
[1]*INRIA Rhone-Alpes, France*
Toan.Nguyen@inrialpes.fr ; lwang@inrialpes.fr
[2]*ALENIA Aeronautica SpA, Corso Marche, Torino, Italy*
vselmin@aeronautica.alenia.it

Abstract

The application designers can usually define their own "virtual environments" by selecting the appropriate computing resources required, or reuse and compose existing environments. The approach is generic by allowing various application domains to benefit from potential hardware and software resources located on remote computing facilities in a simple and intuitive way. The computing resources are defined by services made available as sets of standardized interfaces performing specific tasks: application workflow, input data streams, output visualization tools, monitoring facilities, etc. Services can be composed and hierarchically defined. Transparent access to heterogeneous hardware and software operating systems is guaranteed. An aeroelasticity example in airliner design is given.

Keywords: Multiphysics applications, Virtual environments, Grid computing, Cluster computing.

1. Introduction

The widespread dissemination of grid technology during the last decade has opened new expectations from both the technological and the applications perspectives. The well-known "technology-push/application-pull" paradigm reflects precisely the dynamic equilibrium that state-of-the-art in RTD is seeking. This equilibrium is moving fast, and probably too fast for a majority of potential users.

However, the maturing of grid technology does not suffice to provide acceptable tools for the users. The reserved adoption of Grid technology by the industry today cannot be totally explained by the lack of tools and environments suitable for application design and deployment.

Among the ongoing RTD aspects are the security and Quality of Service items. But more simply is the lack of seamless accessibility tools. Grid technology carries a technical complexity outlook that refrains an important group of its potential users.

Based on these assumptions, this project aims at providing simple tools to allow the application designers and users of the Grid to facilitate its use. It will thus contribute to the uptake of grid technology amongst its huge potential user community.

Building on the claimed virtualization of resources and Virtual Organizations concepts already worked out in the Grid community, the cuurent project proposes to develop user-oriented functionalities for application design, deployment and monitoring on the Grid.

It builds on existing Grid technology to enable application practitioners to seamlessly access and use it. This is far from being the case today, although struggling efforts are currently being undertaken on a large variety of items to mask the technicalities and intricacies of the Grid, in order to make it "transparent". But there is still a long way to go.

Loosely based on the notion of Virtual Organization (VO), this project offers the users of the Grid specific concepts that will facilitate the use of VO. More specifically, the project will implement, deploy and test Virtual Environments (VE), specifically aimed at providing the application designers and users with a dedicated set of tools, protocols and services to support seamlessly their applications in the dynamic grid environments.

Virtual Environments are not specific to any single middleware, but their deployment and testing on realistic applications will be tuned to operational grid infrastructures. They are not specific to any application domain either, and testcases are provided in scientific and business areas. In contrast with VOs, they don't mimic any working organizations: there are no roles, membership and groups in Virtual Environments. They are application-centric, not user-centric services. Their generic nature allows for easy adaptation to specific hardware infrastructures, operating systems software, middleware and applications.

Of particular interest are multidiscipline applications in engineering, as well as environmental monitoring applications and data management application in citizens' life. These will be used as testcases for the validation of the project and will involve both European and Chinese partners.

Because it draws on the existing technology and fast moving grid expertise, this project keeps a close attention to other tools such as the Virtual Workspaces proposed for Globus. It also closely follows the ongoing technology and it uses the WSRF and GT4 as a basis for its implementation. Because interesting research has been carried out on the virtualization concepts, an innovative approach is followed concerning the proposed Virtual Environments.

Indeed, Grid Portals, Grid Application Toolkits, Virtual Workspaces, Dynamic Virtual Environments, Virtual Organizations, Runtime Environments, virtual users accounts, on demand computing and virtual data centers and clusters have all been proposed recently to support the access to the Grid. Much of these important studies and tools have been aimed at single sign-on, authentication, authorization, end-to-end QoS support, and, last but not least, at facilitating the uptake of grid technology by the users. These fundamental bricks are currently being developed, deployed, tested and sometimes marketed on existing infrastructures and middleware.

The approach emphasized here is the cross-leverage of parallel and distributed computing techniques, as supported by grid computing infrastructures, and adequate design and implementation techniques for numerical methods, e.g., domain decomposition, evolutionary algorithms, like genetic and game theory. It is indeed clear that the combined use of several nested levels of parallelism will provide efficient implementations of multidiscipline applications on parallel and distributed infrastructures. An example is given by the French national Grid'5000[5] infrastructure connecting a 100 Itanium2 PC-cluster at INRIA Rhone-Alpes. It is depicted by Figure 1.

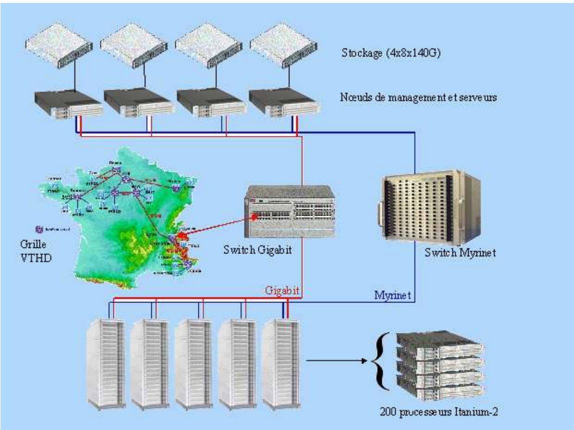

Figure 1. A grid infrastructure.

With the help of some ideas proposed by world renown experts in their field [18], e.g., "virtual flight tests" and "integral operators", we propose in this paper some hints for the deployment of multiphysics code environments on future computing infrastructures.

Our vision is that the ever-growing complexity of computerized environments requires a parallel increase in usability and flexibility of their user interface. Far from the technology barrier hampering the wide dissemination of computing technology in developing countries, there are simultaneously "application pull" drivers that support the demand for ever increasing "technology push" drivers. This non-ending circle has to be made accessible to the application designers and to the end-users, which are not computer science experts.

A simple example is given by the rising tide of grid computing, i.e., the ability to use various computing resources and processors connected by wide-area networks as if it was a single computer for logging, resource reservation, accounting, security, etc. It is currently a technological burden to deploy, use and maintain such environments. The future lies in easy to use, transparent environments, in much the same way that the Internet can be used today by casual users and children altogether, totally unaware of the underlying technologies and infrastructures. An example of such interface developed in the CAST software project at INRIA Rhone-Alpes is given in Figure 2.

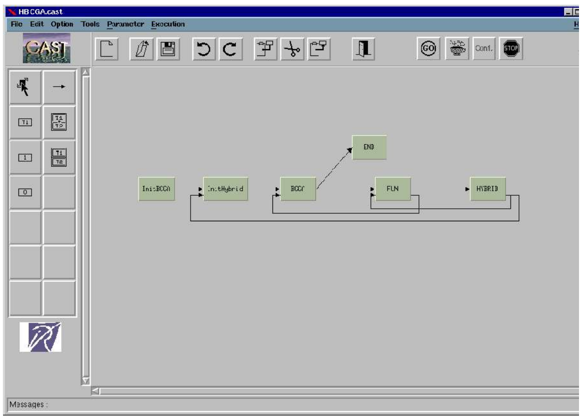

Figure 2. CAST grid-computing interface.

Apart from coupling various disciplines, a net impact of distributed and parallel computing is here the scalability of the problems tackled. Indeed, current processing on large PC-clusters makes available solutions to problems that could not be designed and implemented five years ago.

Because the largest clusters include today thousands of processors[5], grid computing allows the deployment of multidiscipline applications on distributed resources that make such infrastructures a must….

The paper is organized as follows. Virtual environments are introduced in Section 2. Their design and implementation on grid infrastructures is detailed in Section 3. Examples of multidiscipline applications in aerospace design are given in Section 4. Section 5 is a conclusion.

2. Virtual environments

It is a common approach today for network design and deployment to share common physical infrastructures among various logical and possibly overlapping layers of "virtual private networks" [10]. Such an approach bears a number of advantages among which are the ability to scale to the user communities needs, the security which is managed by the underlying infrastructure, the separation of domain addresses, etc.

This is similar to the “distributed virtualization” and “virtual testbeds” developed using overlay networks by the Planetlab consortium [19]. End-user services and “foundational sub-services” are deployed on virtual machines using disruptive technologies in synergistic testbeds and deployment platforms, which are “slices” of the underlying computing and network resources. But this is mainly a network operating systems approach.

A different approach is used here for the design and deployment of “virtual environments” (VE) on grid infrastructures [12].

Here, we focus on application development services rather than network operating systems or end-users services. The goal is to share common computing resources, hardware and software, among various application groups for the secure, scalable and flexible design of application development services.

In contrast with current grid middleware, e.g., Globus, Unicore, the VE do not support directly authentication, authorization, resource brokering and reservation[7, 21]. These are delegated to the grid middleware, which are specifically designed to do that. The added-value of VE is precisely to mask the underlying middleware, in order to simplify access and use of grids. VE enables the straightforward use of application codes without bothering about resource reservation. This is why we call the software layer in charge of VE an “upperware”. Its role is to generate, deploy and manage the VEs.

VEs are sets of possibly overlapping high-level web services deployed by application designers to ease application design and deployment by the service providers. They bear some similarities with “virtual organizations” on grids [4]. But VEs can be implemented on computing environments without grid infrastructures, e.g., on networks of workstations, supercomputers, PCs, etc. They are a generic concept not necessitating grids. VEs are oriented to the application designers’ communities. Their interface is a high-level graphic workflow definition and execution environment [14].

As such, VE are sophisticated service layers building on the “upperware” that in turn relies and uses the middleware functionalities. VE are not just another middleware. Combined with VPN technology, they offer the best of grid computing for the deployment of distributed multidiscipline applications.

3. Implementation

The upperware is a high-level software layer of sophisticated services. Its ultimate goal is to emulate the computing resources, hardware and software, and provide a hosting environment for the applications design and execution, based on powerful computational resources, e.g., grids. In contrast with other approaches specifically aimed at grids [10, 20] the hosting environment can here be any other infrastructure.

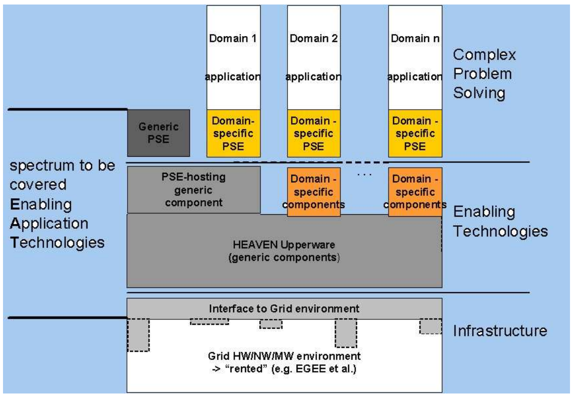

Figure 3. The architecture

Conceptually, it is a hosting infrastructure that supports Virtual Environments emulating hardware and software resources. The environment supports applications that require specific resources. These resources range from computers to storage libraries and sensor devices or visualization and post-processing tools. Virtual environments are isolated from each other, securing the applications from unpredictable application behavior (Figure 3).

The underlying infrastructure ranges from mainframes to wide-area grids of PC-clusters. The current implementation relies on a testbed of several PC-clusters and workstations connected to a high-speed gigabits/sec network [22] (Figure 5).

The CAST[15] application management software is used on top of the Unicore middleware. The details of the testbed infrastructure are given in Table 1.The user interface is an intuitive graphic system which makes the supporting computer technology transparent (Figure 4).

The application components are linked by sequence, parallel and loop operators that from a high-level workflow. An example is detailed in Section 4. This workflow makes transparent all the technical details that are not strictly necessary to the end-users [11]. Components may be implemented in various programming languages, e.g., Fortran, C, C++, or Java. They may be parallel programs involving MPI statements. They may be compliant with CORBA or not, with J2EE containers or not. As such, software components of the application may be grid-aware or not.

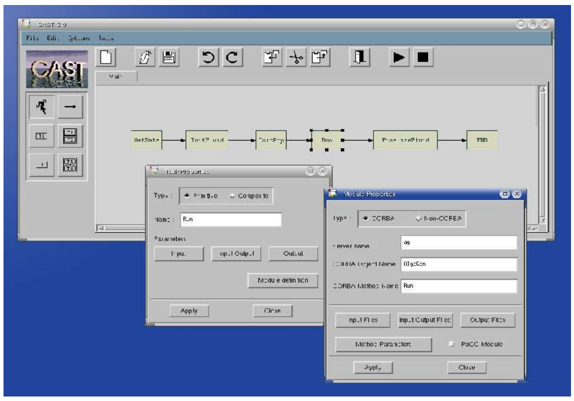
Figure 4. CAST user interface.

Current developments of the upperware will make it compatible with WSDL [1] and WSRF [2] of Globus toolkit G.T.4[7]. This guarantees the compatibility with legacy software and future application software implemented on state-of-the-art middleware.

From a code validation perspective, connections to other devices are necessary. This includes flight-tests results, wind-tunnel experimental data. Ideally, these should be stored in databases for easy access through appropriate Web portals or servers. In this perspective, the connection of the VE to these databases through the underlying infrastructure is straightforward: it is a matter of a few hours. However, performance tuning and assessment have to be established in order not to defeat the speed-up gained by using clusters and high-performance networks. In particular, an everlasting risk is the transmission delays related to the transfers of large volumes of data between application components. It requires the development and use of appropriate transfer protocols [6].

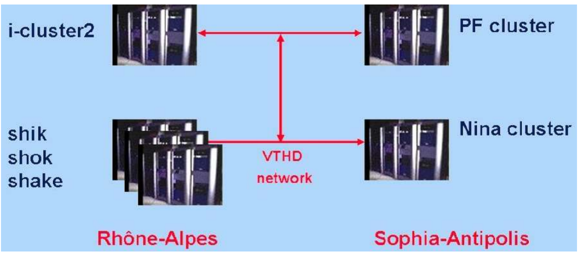

Figure 5. Testbed architecture.

4. Multidiscipline applications

An example of multidiscipline application is presented in this Section, concerning aeroelastic modeling and simulation. It was used in the Promuval project of the EC ("Prospective study on the state of the art of multidisciplinary modelling, simulation and validation in aeronautics" : http://www.cimne.upc.es/PROMUVAL/). The goal was to test CAST as a software integration platform for multidiscipline validation in aeronautics. Major European aircraft manufacturers, together with research centers, where involved in PROMUVAL.

The example was provided by ALENIA Aeronautica (Italy). The goal is to study structural deformations of a medium-size airliner under specific aerodynamic conditions (Figure 6).

It includes the static and dynamic deformations of the wing structure under various load factors, corresponding to different cruise conditions at various speeds and altitudes, as well as pull-up and push-down maneuvers.

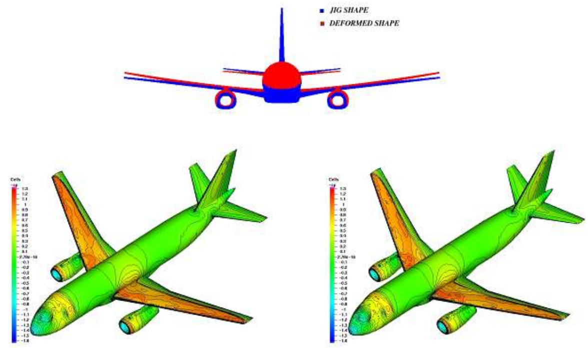

Figure 6. Aeroelastic testcase.

The interactions between the aerodynamics, structural mechanics and mesh generation are described in Figure 8.

The application workflow using the CAST integration platform is depicted Figure 9. It should be noted that although the interactions between the various application components are complex, because they involve a lot of parameters, the design of this workflow is simple. It is basically a loop involving the CFD, CSM and mesh generation/deformation components.

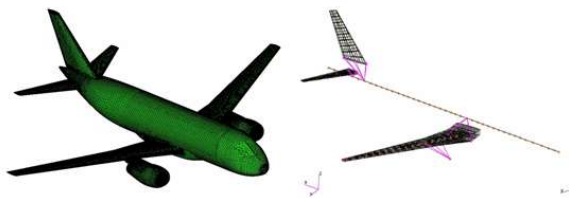
Figure 8. Stuctural model.

This approach also masks the distribution and parallel implementation of the components. The application designers are the only persons to deal with these. This makes the end-users totally unaware of the underlying technical details.

5. Conclusion

We present in this paper an approach for the deployment of collaborative multidiscipline applications on grid computing environments. It is based on a long term cooperation among aerospace and computer science experts from research labs and the industry.

An example provided by ALENIA Aeronautica (Italy) is detailed. It shows the transparent deployment of an aeroelasticity design application on grid infrastructures.

Although focused on collaborative multidiscipline design in aeronautics, it is our belief that our approach is also applicable to many other application areas, ranging from business to industry.

Resource	Hardware & OS	Software
PF @ Sophia Antipolis	Cluster: 19 nodes 100Mbps Fast-Ethernet 1 Node: 2× Pentium III @ 933Mhz Linux Kernel 2.4.2 & LSF	UNICORE server & CAST
NINA @ Sophia Antipolis	Cluster : 16 nodes 3 Gigabit-Ethernet 1 Node: 2× Xeon @ 2Ghz Linux Kernel 2.4.2 & LSF	
i-cluster2 @ Rhône Alpes	Cluster : 100 nodes 1 Gigabit-Ethernet 1 Node: 2 × Itanium @ 900 MHz (64 bits) Linux Red Hat Advanced Server 3.0	UNICORE server
Shok **Shik** **Shake**	Workstation: 1× Pentium III @ 1 GHz 100Mbps Fast-Ethernet Linux Fedora Core 2	CAST, UNICORE client & server

Table 1: Computing resources for the testbed.

Still, the approach presented here requires also the connection to various databases for code execution on relevant enterprise data. This is technically straightforward, but performance evaluation involving such databases remains to be assessed.

Multidiscipline simulation and optimization is the mainstream of research and development for collaborative design in the aerospace industry today. Because it involves several expertise (mathematical modeling, numeric optimization, computer science, e.g., distributed and parallel computing, grid and cluster infrastructures), its deployment in the production arena requires the extensive cooperation of various individuals from multiple teams. To achieve this goal however, a number of barriers still remain. This includes the growing complexity of computing technology, which hampers the easy use of sophisticated computing tools.

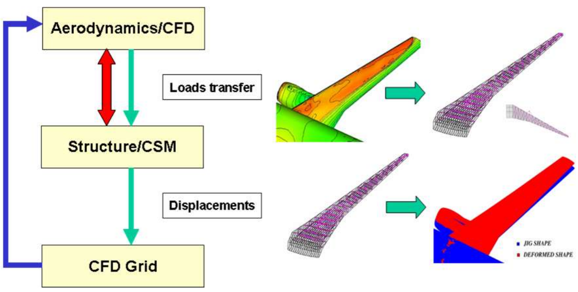

Figure 8. Model interactions.

Commercial services and tools exist today that support "on-demand computing", "utility computing", "user-centric virtual environments" and "virtual datacenters pooling enterprise and partners resources". They usually combine VPN and grid technologies. This is clearly a promising avenue.

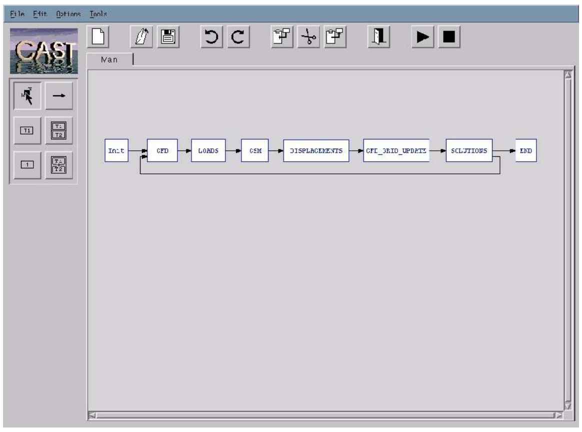

Figure 9. Aeroelasticity application workflow.

In contrast with existing tools that mimic existing enterprise organizations and administrative structures, e.g., Virtual Organizations[22], our approach focuses on the application design, deployment and control. It supports the need for dynamically overlapping resources and complex code-coupling applications. It therefore supports the promising "Virtual Flight Tests"[18] concept.

Acknowledgements

The authors wish to thank Jean-Antoine Desideri, head of the OPALE project at INRIA, for his support and Jacques Périaux (formerly at "Direction de la Prospective", Dassault-Aviation, France) for his long-standing help.

This work was partly funded by the European Commission, FP5 and FP6 "Aeronautics and Space" program, in projects DECISION, FLOWNET, INGENET, MACSInet, PROMUVAL and AEROCHINA.

References

[1]Christensen, E., et al. "The Web Services Definition Language1.1", *W3CNote*, March 2001.http://www.w3.org/TR/2001/NOTE- wsdl 20010315

[2]Czajkowski, K., et al., "The WS Resource Framework", *The Globus Alliance*, March 2004. http://www.globus.org/wsrf/specs/ws-wsrf. pdf

[3] European Community. "The S3 Initiative: a European vision for Software, Services and Systems". Version 31-01-05. January 2005.

[4]Foster, I, Kesselman, C., Tuecke, S. "The anatomy of the Grid: enabling scalable virtual organizations". *International Journal of Supercomputing Applications*, 15 (3). 2001.

[5]Grid5000 http://www.grid5000.org/

[6] "GridFTP: universal data transfer for the grid". *The Globus project. White paper*. The University of Chicago and the University of Southern California. September 2000.

[7] Globus Alliance, "Migrating from GT2 to G4", http://wwwunix.globus.org/toolkit/_docs/development/4.0-drafts/index.html

[8] Grid Interoperability Project, URL: http://www.grid-interoperability.org/

[9] Janka, A., Andreoli, M., Desideri, J.A. "Free form deformation for multilevel 3D parallel optimization in aerodynamics". *International Parallel CFD Conference*. Universita de Las Palmas. Gran Canaria (Spain). May 2004.

[10] Newman, R. "Oxgrid: Virtualisation at Oxford". *Oxford eScience. Campus Grid Workshop*. Edinburgh (UK). June 2005.

[11] Nguyen, G.T., Wang, L., Selmin V. "Virtual private environments for multiphysics applications on computing grids". *Intl' East West High-speed Flow Field Conference*. Beijing (China). October 2005.

[12] Nguyen, G.T., Wang, L., Antikidis, J.P. "Virtual environments for spatial data infrastructures on computing grids". *ISPRS Workshop on "Service and Application of Spatial Data Infrastructures".* Hangzhou (China). October 2005.

[13] Nguyen, G.T., Wang, L., Antikidis, J.P. "Virtual application environments on computing grids". *Third International Conference on Computing, Communications and Control Technologies (CCCT'05)*, Austin, Texas (USA). July 2005.

[14] Nguyen, G.T., Antikidis, J.P. "Virtual computing environments for problem-solving on grids". *International Parallel CFD Conference*. Universita de Las Palmas. Gran Canaria (Spain). May 2004.

[15] Nguyen, G.T. "Grid-computing in Multidisciplinary CFD optimization problems". *Invited lecture. International Parallel CFD Conference.* Russian Academy of Sciences. Institute of Mathematical Modeling. Moscow (Russia). May 2003.

[16] Nguyen, G.T. "Integration of multidiscipline applications in grid-computing environments". *Invited lecture . PARA02 Conference on Applied Parallel Computing*. Center for Scientific Computing. Helsinki (Finland). June 2002.

[17] Nguyen, G.T. "An Integration platform for metacomputing applications". *International Conference on Computational Science ICCS 2002*. Amsterdam (NL). April 2002.

[18] Perrier, P., "Virtual flight tests", *Promuval Workshop: Prospective in multidisciplinary modeling, simulation and validation in aeronautics.* National Technical University of Athens (Gr). November 2004.

[19] PLANETLAB http://www.planet-lab.org/

[20] Ruth, P., Jiang, X., Goasguen, S. "Virtual distributed environments in a shared infrastructure". *IEEE Computer*. May 2005.

[21] UNICORE project, www.unicore.org

[22] Denemark, J, et al. User Management for Virtual Organizations. CoreGRID Technical Report TR-0012. Novmber 2005.

[23] Keahey K., Doering K., Foster I. From sandbox to playground: dynamic virtual environments in the grid. Proc. 5th IEEE/ACM Int'l Workshop on Grid Computing. GRID'04. Pittsburgh (USA). November 2004.

[24] Keahey K., Ripeanu M. , Doeirng K. Dynamic creation and management of Runtime Environments on he Grid. Workshop on Designing and Building Web Servces GGF9. Chicago (USA). October 2003.

[25] Kehaey K., et al. Virtual Workspaces in the Grid. Europar 2005. Pisa (I). August 2005.

[26] Foster I. , et al. Modeling stateful resources with Web Services. Versin 1.1. March 2004.

[27] Jankowski M., Wolniewicz P., Meyer N. Virtual user System for Globus based grids. Proc. Cracow Grid Workshop December 2004.

[28] Chase J., et al. Dynamic virtual clusters in a grid site manager. Proc. 12th Int'l Symposium on High-performance Distributed computing. 2003.

[29] Chuang Lin, Zhiguang Shan. Grid research in China: origins and directions. Proc. 10th IEEE Int'l Workshop on Future Trends in Distributed Computing Systems. 2004.

[30] Laszewski L., et al. Commodity Grid Kits: Middleware for building grid computing environments. Chapter in "Making the Global infrastructure a Reality". Wiley Eds. 2003.

[31] Lorch M., et al. Authorization and account management in the Open Science Grid. Proc. IEEE Grid Computng Workshop. 2005.

[32] NESSI. Networked European Software and Services Initiative. Vision Document. Version 1.2b. May 2005.

[33] Davis K. Grid Application Toolkit Canonical Implementation and user's guide. GridLab project. CGUG-0013-1.0