Distributed Virtual Test Bed: an Approach to Intergation of CAE Systems in UNICORE Grid Environment

G.Radchenko and E. Hudyakova
South Ural State University, Chelyabinsk, Russia
radchenko@acm.org

Abstract - Computer-Aided Engineering (CAE) systems demand a vast amount of computing resources to simulate modern hi-tech products. In this paper, we consider the problem-oriented approach to access remote distributed supercomputer resources using the concept of distributed virtual test bed (DiVTB). DiVTB provides a problem-oriented user interface to distributed computing resources within the grid, online launch of CAE simulation, automated search, monitoring and allocation of computing resources for carrying out the virtual experiments. To support the concept of DiVTB the DiVTB technology was developed. DiVTB technology provides a solution for the development and deployment of DiVTB, integration of most common CAE systems into the distributed computing environment as grid services (based on the UNICORE grid middleware) and web access to CAE simulation process.

I. Introduction

Computer-Aided Engineering systems are now one of the key factors ensuring competitiveness of any high-tech production. The use of such systems makes it possible to conduct virtual experiments when real-life experiments are difficult or impossible. This can greatly improve the accuracy of case analysis and design decisions and shorten the way from the generation of an idea to the real industrial production [7].

The accuracy of the computer simulation depends on the level of detail of meshes used for computational experiments. The computational complexity of problems of engineering analysis keeps increasing, and their solutions require significant computing resources. The only way is to use the multiprocessor systems.

For an ordinary user the process of solution of engineering problem involving supercomputing resources is associated with certain difficulties. First, the interface and features of all software components that provide the simulation technological cycle need to be explored [2]. Second, specific knowledge and skills in the high performance computing are required. And finally yet importantly, the cost of the purchase, installation and administration of powerful supercomputer equipment and software is extremely high. All these factors complicate the process of embedding of powerful engineering methods in research and development in traditional manufacturing companies.

An alternative to creating your own supercomputer center is renting remote computing resources from the supercomputer centers that offer utility computing services. The utility computing assumes that diverse computational resources can be brought together on demand and that computations can be implemented depending on demand and service load [6]. However, there is a range of issues related to the mechanism of remote resources usage. Issues of data exchange security, transparency of allocation and use of remote resources can deter industrial users from such approach. These issues can be resolved by the grid computing concept [4] together with cloud computing concept [5], whereby the end user has a problem-oriented service that provides problem solution based on distributed computing resources.

In the past few years, the approach to supplying the CAE systems in the form of distributed computing environment resources is booming. In [3, 11] development of collaborative design system for engineering analysis by dynamically distributed development team is introduced. As members of such groups need to interact constantly during the development process, peer-to-peer computing was selected as the basic platform for this system. However, the use of this approach is not aimed at providing the end user with a finalized product accessing distributed network resources, but rather at supporting the collaborative design process for a development team.

In [9] a concept of the Virtual Test Bed (VTB) (a prototype of the virtual cooperative engineering environment) was introduced. The VTB has been designed as an architecture to facilitate the integrated execution of different simulation programs with other supported non-simulation software. The proposed solution is built on the Extensible Modeling and Simulation Framework (XMSF) platform that supports such distributed computing approaches as CORBA, SOAP, DCOM and is highly targeted at the specific field of application (the development of new generation spacecrafts).

In [1] the REST web-services architecture for distributed simulation is introduced. The proposed approach supports interoperability of collaboration of

The reported study was partially supported by RFBR, research project No. 11-07-00478-a and by the Ministry of Education and Science of Russia, task No. 8.3786.2011.

distributed computing resources and robust data exchange during simulation process.

The analysis of virtual test beds for CAE systems shows:

- the lack of a unified approach for creating virtual test beds allowing CAE simulation in the distributed computing environment on the basis of standard grid protocols;
- the lack of adaptation of the VTB problem-oriented user interface for the needs of the certain categories of users and the lack of streamlined web interface for inputting the tasks;
- lack of a problem-oriented resource broker, which would transparently provide resources of distributed computing environment (including hardware, software and license resources) in compliance with the current task requirements.

In [8] the concept of GridBean in GPE grid middleware platform was introduced. A GridBean is an object responsible for generating the job description for grid services and providing graphical user interface for input and output data. This platform introduced an easy way to integrate legacy stand-alone applications in the grid environment as grid services. Subsequently the GridBean concept was integrated into the UNICORE middleware platform [10].

In this paper, we provide a review of the Distributed Virtual Test Bed (DiVTB) concept that allows providing resources of CAE systems in the grid computing environment. Moreover, we present the DiVTB Technology — a combination of concepts and techniques used for the development and operation of DiVTB. The UNICORE grid middleware is used to implement grid services that integrate CAE systems in grid environment.

II. The Distributed Virtual Test Bed

A. Basic terms

Let a CAE-problem be assumed as the combination of:

- a geometric model of a research target and/or computational mesh which divides a simulated area into discrete sub-areas;
- boundary conditions, physical characteristics and parameters of the components interaction in a simulated area;
- a description of unknown variables which values will be obtained as a result;
- the requirements for hardware and software resources that provide a simulation process.

Let a CAE-parameter be the value, which affects the result of simulation and can vary within a certain range. Each parameter has specific semantics and describes a feature of a subject area or a solution process.

We offer a concept of a Distributed Virtual Test Bed (DiVTB) to provide access to resources of CAE systems in a distributed computing environment. DiVTB is a software solution that provides an engineering simulation in the grid computing environment for a certain CAE-problem. DiVTB includes an interface for CAE-parameters values input; a set of software tools enabling the use of grid resources for virtual experiment (*a driver*) and a set of grid services that provide the solution of a CAE-problem and implement safe standardized remote communication methods.

A combination of concepts and techniques for DiVTB development and operation is defined as *the DiVTB Technology*.

The DiVTB Technology includes:

- the DiVTB architecture and development techniques;
- the administrative concepts: the patterns of work and the
- distribution of duties between the DiVTB developers and users;
- the software solution for the DiVTB development, deployment and operation.

B. The structure of a DiVTB

CAEBean is a primary structural unit of a DiVTB. According to the DiVTB Technology, we distinguish four structural levels of a DiVTB, each represented by its CAEBean type (Figure 1):

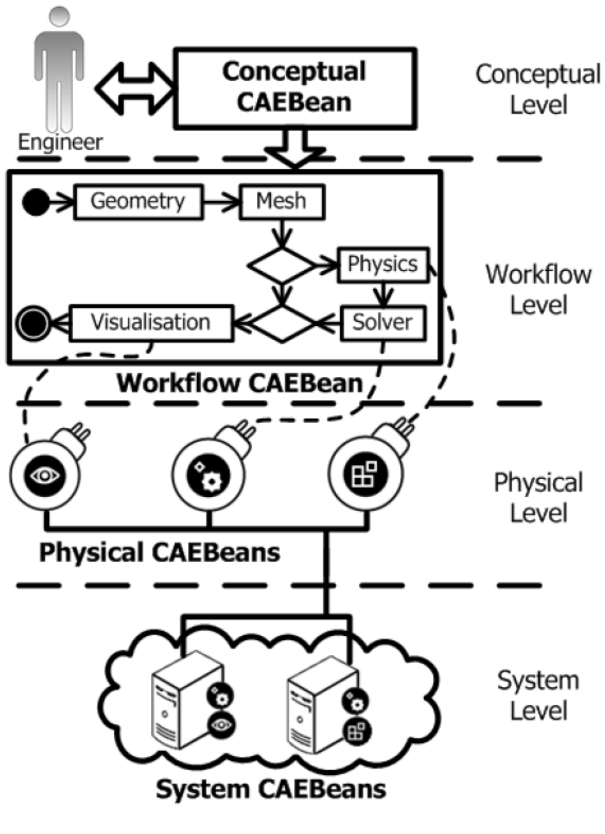

Figure 1. A generalized scheme of DiVTB levels according to the DiVTB Technnology

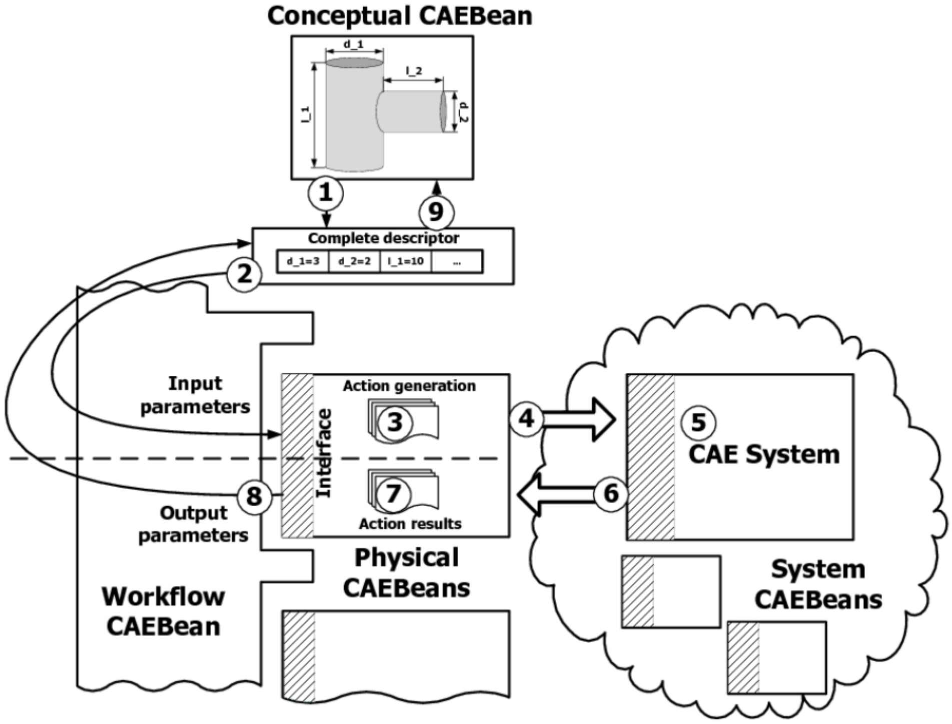

Figure 2. Simulation of a distributed virtual test bed

- Conceptual level (a hierarchy of *conceptual CAEBeans*);
- Workflow level (a *workflow CAEBean*);
- Physical level (consists of a set of *physical CAEBeans*);
- System level (consists of a set of *system CAEBeans*).

A conceptual level of DiVTB is based on conceptual CAEBeans. A conceptual CAEBean defines the *problem-oriented interface* for a DiVTB based on a set of its CAE-parameters. Through a conceptual CAEBean a user can launch the DiVTB simulation, follow a solution process and obtain the results. The DiVTB Technology provides the opportunity to design a *generalization hierarchy* of conceptual CAEBeans. The conceptual CAEBeans of lower hierarchy levels adapt a problem-oriented interface of the DiVTB for the needs of a certain category of users. These CAEBeans specify the CAE-problem defined by a parent conceptual CAEBean through assigning specified values to some CAE-parameters.

The workflow level of DiVTB is represented by a workflow CAEBean implementing a CAE-workflow for a certain CAE-problem. Let a CAE-workflow be defined as a directed graph with the following types of nodes:

- an action performed by a certain CAE system;
- a node controlling the problem solution flow.

We use the elements from the UML 2.0 activity diagram notation to form a CAE-workflow. The information required to initiate any node of a workflow is stored in a *complete descriptor* of DiVTB. Let a complete descriptor be defined as a set of CAE-parameters that uniquely describes a CAE-problem, which can be simulated by DiVTB. Any workflow node as a part of a solution can refer to a *complete descriptor* for input values and store results in the corresponding output parameters.

A physical level of DiVTB is represented by the physical CAEBeans. The main function of a physical CAEBean is a conversion of a problem-oriented description of a CAE-problem (a set of parameters in complete descriptor) into the set of files needed to launch a single *CAE-action* (mesh generation, simulation, post processing, optimization etc.) on a specific CAE system. At the end of a problem-solving process a physical CAEBean provides conversion of component-oriented results in a problem-oriented form. During the CAE-workflow execution, each physical CAEBean must be matched with a grid service to execute a CAE-action. All grid services are implemented through the DiVTB System. A workflow CAEBean together with the corresponding physical CAEBeans implement *a driver* of DiVTB.

A system CAEBean provides the functionality of a computing resource in a distributed computing environment and implements a service-oriented approach to task input and fetching the results. A system CAEBean provides the isolated workspace for each CAE-action and the programming interface that allows initial data loading, remote task input and delivery of results.

C. *DiVTB Simulation*

Let us consider the process of DiVTB simulation according to the DiVTB Technology (Figure 2).

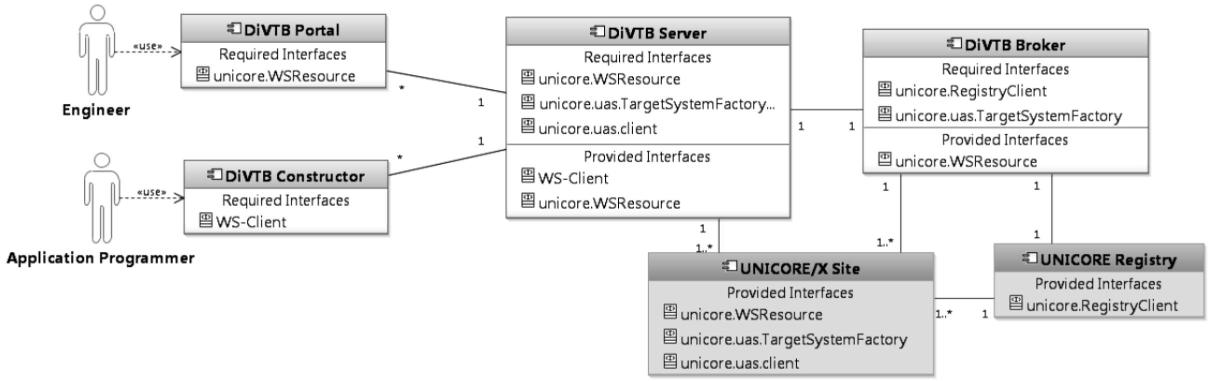

Figure 3. General diagram of DiVTB System components

1. An engineer inputs the values of CAE-parameters for DiVTB by means of user interface provided by an appropriate conceptual CAEBean. The conceptual CAEBean calculates values of CAE-parameters, which are not available to the user on the current level of generalization hierarchy. All initial CAE parameter values are stored in the complete descriptor of DiVTB. Finally, the engineer launches the simulation process.
2. The workflow CAEBean executes the CAE-actions according to the CAE-workflow. The values of initial parameters of any action (to be defined before the action starts) are taken from the complete descriptor of DiVTB and transferred to the physical CAEBean.
3. The physical CAEBean generates a set of input files for the CAE system wrapped in the corresponding system CAEBean.
4. The resource broker gives an address of the CAEBean system with the best opportunity to execute the CAE-action. The physical CAEBean transfers a set of input files to the allocated CAEBean system and starts the CAE-action.
5. The CAE system executes the CAE-action and generates a set of result files.
6. The Physical CAEBean receives result files from the system CAEBean.
7. The physical CAEBean parses the results of the action and retrieves the values of required output parameters.
8. The physical CAEBean puts the obtained values of output parameters into the complete descriptor. The workflow CAEBean triggers the next node of the CAE-workflow (step 2).
9. When the CAE-workflow is complete, all results of the DiVTB simulation are transferred to the conceptual CAEBean from the complete descriptor.

III. THE DIVTB SYSTEM

To implement the DiVTB Technology we created a DiVTB System — a software solution for DiVTB development and implementation. We selected the UNICORE 6 grid computing middleware [10] as a platform for implementation of grid services. The GridBeans approach included in UNICORE 6 supports the transparent integration of legacy standalone applications as grid services in the grid environment. The DiVTB System consists of the following components (Figure 3):

1. *DiVTB Constructor* — integrated development environment of DiVTB. It provides the interface for an application programmer to develop the CAEBeans of conceptual, workflow and physical levels. Accordingly, the DiVTB Constructor user interface is divided into three sections responsible for developing corresponding types of CAEBeans.
2. *DiVTB Portal* — a web application that provides a user interface for the input and solution of CAE-problems by means of an appropriate DiVTB. The web-form generator automatically builds DiVTB user interfaces based on the corresponding conceptual CAEBeans. In addition, the DiVTB Portal supports authentication and user account management.
3. *DiVTB Server* — a grid service for DiVTB storage and simulation. A process of execution of a CAE-workflow is supported by the internal workflow subsystem of the DiVTB Server. DiVTB Server provides interfaces for DiVTB Constructor and DiVTB Portal allowing to upload a DiVTB to a Server and to start a simulation of a DiVTB.
4. *DiVTB Broker* — an automated system for registration, analysis and allocation of CAE-resources for CAE-action execution. The DiVTB Broker receives requests for resources from DiVTB Server, analyzes the status of the distributed computing environment and supplies the DiVTB Server with a CAE-resource best suited to the request. DiVTB Broker interacts with an UNICORE Registry service to gain an information about resources available in the grid environment.

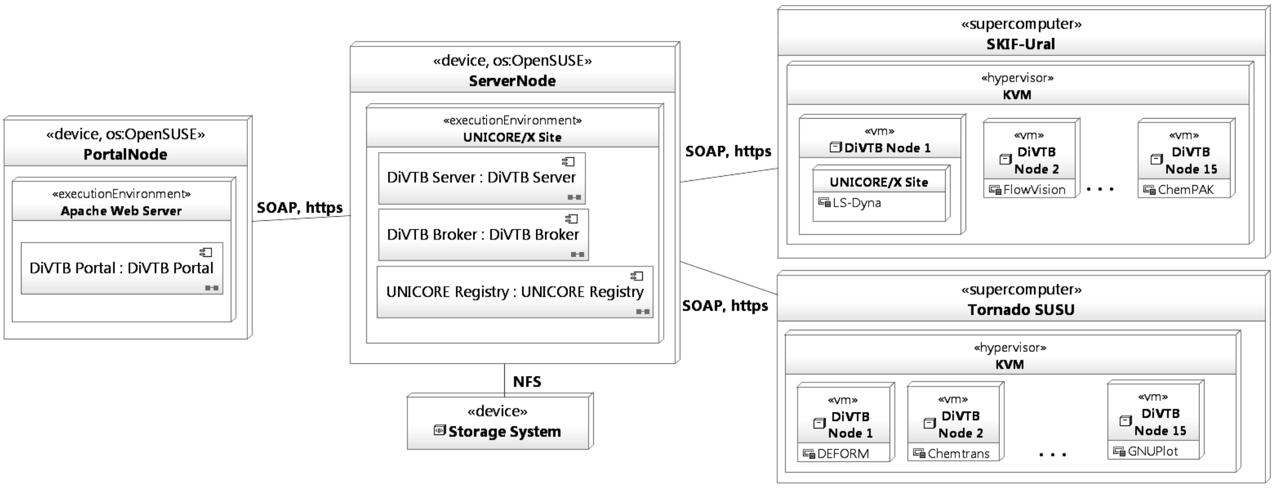

Figure 4. DiVTB System deployment diagram

5. *CAE-Resources* — grid services that implement the remote execution of CAE-Actions by means of CAE-systems installed on specific computers in a grid (instances of system CAEBeans). Each CAE-resource is a UNICORE/X site with a set of problem-oriented and system software installed on it. CAE-resource provides:

- obtaining problem data from DiVTB Server or an external data source;
- CAE-action initiation and automated solution;
- transmission of results into the DiVTB Server or external data storage.

IV. DiVTB System Test

To test the DiVTB System we developed a set of DiVTB to solve problems by using common CAE systems: ANSYS CFX, ANSYS Mechanical, ABAQUS, DEFORM, LS-DYNA. In addition, a set of self-written and open-source CAE components was integrated in DiVTB System during "Young Researchers Mobility" research project, which was held at the South Ural State University in autumn 2012. DiVTB System was deployed on the supercomputer resources of South Ural State University Supercomputer Simulation Laboratory (Figure 4).

For the test we chose the issue of tempering and cooling of tubes and analysis of impact of various aspects of the quenching process on the quality of the production (*"Thermal Treatment Distributed Virtual Test Bed"*) based on DEFORM system.

Firstly, we created a model of a heat treatment process. We supplied thermal studies of the pipe hardening. We collected information about temperature fields during the quenching process and the magnitude of the initial circumference and curvature along the axis of the work piece. The "Thermal Treatment" DiVTB allows to change the flowing parameters of the heat treatment process: number of inductors, frequency and current strength, a length of the inductors, number and configuration of the jets of water, pressure of cooling flow, velocity of pipe motion, pipe steel grades (Figure 5).

Another application of DiVTB was a virtual test bed for supercomputer simulation of deformation of knitted fabrics on human body. We designed and deployed a DiVTB based on LS-DYNA CAE system that allow simulating the process of putting on knitted clothes to a torso of a human body. By varying parameters, such as density and method of weaving of knitted material, engineer can analyze the quality of the fit of the final product; view a video of the process of dressing, etc.

Testing revealed that end-users easily adapt to the solution of applied problems on remote computer infrustructures through a DiVTB System. Web-based user interface provides seamless access to grid resources without the need for "classical" techniques for working with remote and supercomputer resources, such as SSH console access to and SFTP. Advanced users also prefer to develop a simplified DiVTB to organize their everyday use of remote supercomputing resources.

In addition, the tests revealed a number of features that should be considered in the further development of DiVTB System.

First, communication between developers and end-users of DiVTB System often held directly at the workplace of the customer. In this case, the use of a DiVTB Constructor desktop application complicates the development process and interaction with the customer.

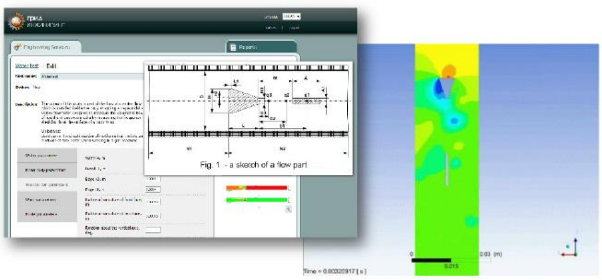

Figure 5. Example of DiVTB user interface (DiVTB Portal)

To solve this problem, we create a Web application that provides DiVTB development.

Secondly, it appears that the most challenging in terms of Application Programmer became a procedure of templating of the problem and result files (Action Generation and Action Results stages on fig. 2). Application Programmer has to create custom stand-alone applications that provide the implementation of these procedures, although in most cases he can use rather simple text generators based on certain templates. To solve this problem, we are planning to develop a DiVTB Constructor module that will provide template generation and customization of input data.

In addition, users asked us to provide pre-visualization of simulation results before loading it on a user's machine. Since the simulation results can be as big as hundreds of gigabytes, downloading this amount of data from a remote HPC system to a user's machine can take a very long period. If the results do not meet user's expectations, he would like to know about it in advance. A possible solution to this problem is to provide a remote interactive visualization of calculation results, so user can visualize simulation data on the site of its calculation. To provide this capability, we are developing a remote interactive visualization service that would provide such capabilities.

V. CONCLUSIONS

This paper presents the main aspects of CAE systems integration in the distributed computing environment. We have offered the concept of distributed virtual test bed (DiVTB) providing a problem-oriented approach for CAE-problems simulation by means of grid computing resources. We have developed a DiVTB Technology allowing development and operation of distributed virtual test beds. Based on this technology we have developed a DiVTB System and deployed virtual test beds aimed at solving CAE-problems by means of the most common CAE systems (ANSYS CFX, ANSYS Mechanical, ABAQUS, DEFORM, LS-DYNA).

REFERENCES

[1] K. Al-Zoubi and G. Wainer, “Using REST web-services architecture for distributed simulation,” 2009 ACM/IEEE/SCS 23rd Workshop on Principles of Advanced and Distributed Simulation, PADS ’09, Washington, DC, USA, pp 114–121, June 2009.

[2] T. M. Buriol and S. Scheer, “CAD and CAE integration through scientific visualization techniques for illumination design,” Tsinghua Science & Technology, vol. 13, no. 1, pp. 26 – 33, October 2008.

[3] L. Q. Fan, A. Senthil Kumar, B. N. Jagdish, and S. H. Bok, “Development of a distributed collaborative design framework within peer-to-peer environment,” Comput. Aided Des., vol. 40, no. 9, p. 891–904, Sept. 2008.

[4] I. Foster and C. Kesselman, The Grid 2: Blueprint for a New Computing Infrastructure. The Morgan Kaufmann Series in Computer Architecture and Design Series. Elsevier Science, 2003.

[5] B. Hayes, “Cloud computing,” Commun. ACM, vol 51, no 7, pp. 9–11, July 2008.

[6] M. Huhns and M. Singh, “Service-oriented computing: key concepts and principles,” Internet Computing, IEEE, vol. 9, no. 1, pp. 75 – 81, Feb. 2005.

[7] B. Raphael and I. Smith, Fundamentals of Computer-Aided Engineering. John Wiley & Sons, 2003.

[8] R. Ratering, A. Lukichev, M. Riedel, et al. “GridBeans: support e-science and grid applications,” In Proceedings of the Second IEEE International Conference on e-Science and Grid Computing, E-SCIENCE ’06, p. 45, Washington, DC, USA, 2006. IEEE Computer Society.

[9] J. Sep_ulveda, L. Rabelo, J. Park, F. Riddick, and C. Peaden, “Implementing the high level architecture in the virtual test bed,” proceedings of the 36th conference on Winter simulation, pp. 1444-1451, 2004.

[10] A. Streit, P. Bala, A. Beck-Ratzka, et al, “Unicore 6 - recent and future advancements,” Annals of Telecommunications, vol 65, pp. 757-762, 2010.

[11] J. Yin, W. Zhang, Y. Li, and H. Chen, “A peer-to-peer-based multi-agent framework for decentralized grid workow management in collaborative design,” The International Journal of Advanced Manufacturing Technology, vol. 41, pp. 407-420, 2009.