IMPLEMENTATION OF FINITE ELEMENT DISTANCE LEARNING AND RESEARCH TOOLS BY USING WEB SERVICES

Ramute Barauskiene, Rimantas Barauskas, Vytautas Daniulaitis

Kaunas University of Technology, System Analysis Department, Studentu 50, Kaunas, Lithuania, ramute.barauskiene@ktu.lt, rimantas.barauskas@ktu.lt, vytautas.daniulaitis@ktu.lt

Abstract. A structure for finite element analysis distance learning, numerical laboratory examples, course material and communications scheme has been proposed in order to deploy them in distributed resources of GRID computer network. The application cases of activities of the virtual organization have been identified. Two implementation versions of web applications have been realized in the form of HTML pages and in the form of Java-XML web services. The paper represents current stage of implementation of the task "Learning tools of FEA" carried out in the scope of the ESF project "Creation, development and implementation of virtual science and studies organization on the base of GRID technologies".

Keywords: finite element software, distance learning, Web services

1 Introduction

Finite element analysis (FEA) is the most popular techniques for presenting the behavior of physical systems in "virtual reality". At present time it is commonly realized as a technology of the physically based modeling rather than the numerical finite element method, which comprises the mathematical kernel of the technology. The FEA software involving adaptive meshes, multi-level solvers, parallel and distributed computing features are rather complex and difficult to interfere and to modify. Generally the finite element modeling is close to a professional activity based on a deep knowledge of the FEA mathematical background, thorough understanding of the engineering essence of the problem, as well as, experience in using FEA software.

The FEA is a fundamental mathematical and engineering learning subject, which is classified under Computational Science and Engineering (CSE) studies program. It is oriented for computational analysis of physical phenomena and engineering solutions and is based on computer science, applied mathematics and on engineering knowledge of the application area. The distinguishing feature of education in FEA lies in its interdisciplinary nature, which often requires continuous and life-long education of students and professionals of very different professions and occupations. Creating possibilities for distance learning and research in the field of FEA may facilitate the studies process. The access of studies materials and software via Internet enable the distant students to submit batch jobs for the up-to-date FEA software (e.g. ANSYS, LS-DYNA, MSC, COMSOL), which are licensed on appropriate servers and usually are not available on personal computers of individual students. Even in the case of availability of free licenses for students, the access to distant powerful computers facilitates the solution of time-consuming computational jobs. Deeper reasoning for creation of virtual learning and research tools has been presented in [1,2]. The systematic development of web applications (WA), which enabled FEA studies process and practical work materials over Internet at Kaunas University of technology has begun since 2006 during the implementation of project [3].

2 Application cases of FEA learning and research services

Learning-oriented FEA software practically is the same as used for solution of real practical problems. Numerous solvers, software libraries and versions oriented for parallel computers are available on the market. Among them programs ANSYS [4], LS-DYNA [5], which possess their own user interfaces, mathematical computations environments MATLAB + COMSOL Multiphysics [6] allow high level programming by using MATLAB scripting language. Many FEA tasks can be solved by writing user programs in high-level programming languages C+, C#, Java, FORTRAN, etc.

In process of FEA studies many students, post-graduates ad teachers communicate by using the common software in order to fulfill their own goals. Many participants of the process are eager to share the FEA software and computational resources. The sharing should be understood as possibility of the direct access to the software, data and other resources residing on distant computers. The suppliers and users of the resources define the object and conditions of sharing. The group of individuals and/or institutions, the interaction among which is defined by sharing rules is defined as the *virtual organization* (VO) [7].

The proposed structure of application cases (AC) of a learning and research VO in the field of FEA reads as follows.

***AC1*: *FEA by using commercial FE software*.** The users present their tasks in the form of programs-scripts, which are written in the internal language of the particular software system(e.g., language APDL of

ANSYS, “keyword” files of LS-DYNA, etc). The web applications enable the execution of scripts on distant powerful parallel computers or clusters, as well as, transfer of input data and results files.

AC2*: *FEA by performing computations in mathematical computations environments (MATLAB+COMSOL Multiphysics). This way of problem solution is reasonable in the cases of “non-standard” problems, where the usage of commercial software may appear inconvenient or not suitable. On the other hand, this approach is very useful during the FEA learning process. The student may write programs implementing one or several stages of the computation, while remaining stages can be implemented by using the existing programs or functions. Two levels of creating a FEA application are identified:

- *Basic functions level:* element matrix functions, structural matrix assembling functions, constraint functions, material model(constitutive equation) functions, etc.;
- *Computational strategy level:* iterative schemes for non-linear problems, semi-analytical approaches, structural non-integrity modeling, etc.

***AC3*:** *Application of numerical and graphical libraries* is closely related with tasks solved in AC2. A lot of accompanying software may be used during implementation of real problems FEA solutions, e.g., sparse matrices, signal processing, image processing, results visualization, etc.

***AC4*:** *Distance learning: access to e-books, e-guides and examples* may be treated as conventional distance learning tools. However, we also consider the possibility of “active examples”, where the student is able to enter the parameters describing the particular example, to obtain and analyze the results. Implementation of such services is based on the distant program invocation for performing the solution of the example and in this aspect correlates with AC2

The users are grouped as:

“Students” – persons pursuing the aim to acquire the basic or advanced level of education in the field of FEA. Their main tasks are to study the course materials and/or to perform the tasks stinted by the teacher. The students participate in AC2 and AC4. or sometimes they can participate in AC1 as users with restricted(limited) resource access.

“*Post-graduates”* and *“researchers”* comprise two similar users groups interested in solution of advanced practical problems by using FEA software. The two groups are different in assigning them appropriate licensed computational resources. Both groups have established rights to participate in all above mentioned application cases.

3 Implementation of Web Applications

Techniques and technology of implementation of Internet access to information and resources is a rapidly developing field, continuously acquiring new aspects and features. All available means of creation of active web-pages and applications, such as HTML(XHTML) pages combined with browser scripting (JavaScript) and server scripting (PHP) capabilities, Java applets, Java servlets, Java Server Pages, JAXWS, etc. can be applied for fulfilling the task to create the functionality mentioned in above described Application Cases of the virtual organization. GRID applications introduce their own requirements, the main point being the statefullness of the created WA. This means creating the virtual resource for the client and remembering its state during the session, as well as, in-between two sessions with this particular client.

<u>Reusability</u> of FEA distance learning and research services created by using different technological platforms appears to be an important issue in order to maintain the WA convenient and attractive for the user. On he other hand, this could reduce as much as possible the intellectual effort necessary to include the earlier created product into newer environment. The prototypes of WA created by means of HTML+JavaScript+PHP are useful as they can be run on the local network immediately and be tested in conditions of real usage. The real GRID applications can be created by using the Globus Toolkit 4 (GT4) system [8]. However, the present version of the system had no tools for automatic creation of XML-resources, which could enable proper automatic deployment of the WA. In this work we used the NetBeans5.5 environment for creating the “intermediate” WA with full required functionality, the XML resources of which are generated automatically. Further they are extended to XML descriptions necessary for deployment of the WA in GT4.

3.1 Main features of WA implemented by using HTML+JavaScript+PHP

Two WA have been created, implementing VO application cases AC2 and AC4. They have the form of HTML web pages, which contain fields for entering the necessary user information, activation buttons for necessary programs to be run on the server and HTML references for viewing the computation results. They have the appearance of active web pages, as they inform the user immediately about the actions made by the used program by means of information fields and/or windows appearing on the client window. The hearth of both created WA is the remote activation and control of the **MATLAB** program, one or several instances of which reside on the server computer and are driven by the <**`matlabserver`**> program supplied together with

MATLAB webservices toolbox. The functions of the client and server are explained in Figure 1, which presents a simplified scheme of WA **BEM_pvz_1**(“Run FEA example in MATLAB”). The client posts the HTML form data by invoking the action <**action=matweb.exe**>. There is a hidden field, called <**instruct.mlmfile**> that references the particular function M-file **BEM_pvz_1.m** , which is prepared in accordance with the template

function retsr=BEM_pvz_1(instruct, outfile)

in the case of WA represented in Figure 1.

The <**matweb.exe**> program is invoked on the server. It reads the form data posted by the client on the server port 8080 and creates the <**instruct**> structure. <**instruct**> contains fields corresponding to the HTML form fields in the HTML form. The variables of <**instruct**> structure (e.g., **f1, f2**,…) can be directly used in the **BEM_pvz_1** function body, which contains also a call to the pre-prepared **MATLAB** script **pvz11**. **pvz11** describes the actual flow of the statements used for solving the example. Further **BEM_pvz_1** function is sent by <matweb.exe> to the running **MATLAB** instance via port 8888 for execution. It is also <matweb.exe>, which receives the log file data from **MATLAB** output and sends it via the server port 8080 to the client. The instance of **MATLAB** creates necessary results files (e.g., picture

BEM_pvz_1_rez.jpeg), which can be accessed by the client via appropriate reference

<a href=BEM_pvz_1_rez.jpeg … >.

The client-server interaction is driven by Apache http server.

The <**matweb.exe**> is implemented on the base of the obsolescent Common Gateway Interface (CGI) protocol defining the server interaction with the programs, which are running on client computers. Though the practical implementation of WA is quite convenient and easy, the further use of such communication protocol for FEA learning and research WA is complicated because of the fact that new versions of Internet browsers are blocking the direct invocation of executables. As a consequence, **<action=matweb.exe>** HTML directive does not work reliably any more. We did not continue further WA creation on this basis any more. Instead, new developments have been made on the base of Java-based WA descriptions. Fortunately, many HTML pages and prepared MATLAB scripts could be reused in the new context.

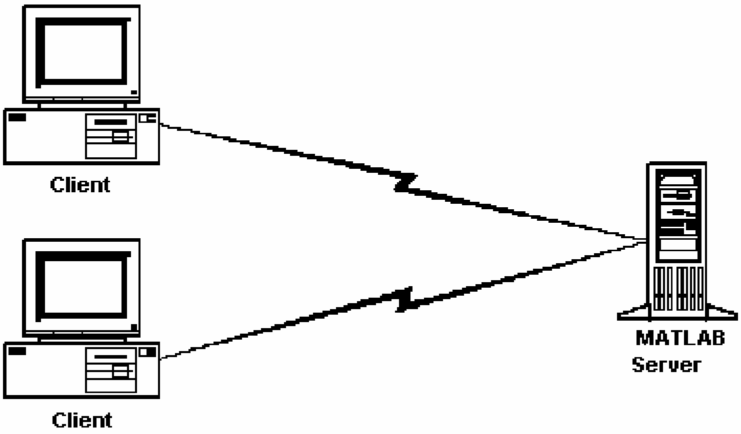
3.2 Main features of a web applications implemented by using Java-XML web services

A WA implementing VO application case AC4 has been created. Client requests and web service responses are transmitted as Simple Object Access Protocol (SOAP) messages over HTTP thus enabling a completely interoperable exchange between clients and web services, all running on different platforms and at various locations on the Internet. HTTP is a request-and response standard for sending messages over the Internet, and SOAP is an XML-based protocol that follows the HTTP request-and-response model [9].

The SOAP portion of a transported message handles the following:

- Defines an XML-based envelope to describe what is in the message and how to process the message
- Includes XML-based encoding rules to express instances of application defined data types within the message
- Defines an XML-based convention for representing the request to the remote service and the resulting response

The Web Services Description Language (WSDL) is a standardized XML format for describing network services. The description includes the name of the service, the location of the service, and ways to communicate with the service. By using NetBeans5.5 environment for creating the web service and client programs, the five necessary web-resource files (manifest.mf, web.xml, context.xml, *.wsdl and *.xsd) are generated automatically. The general communication scheme among the client portlet (or browser window), web client program and web service program is presented in Figure 2.



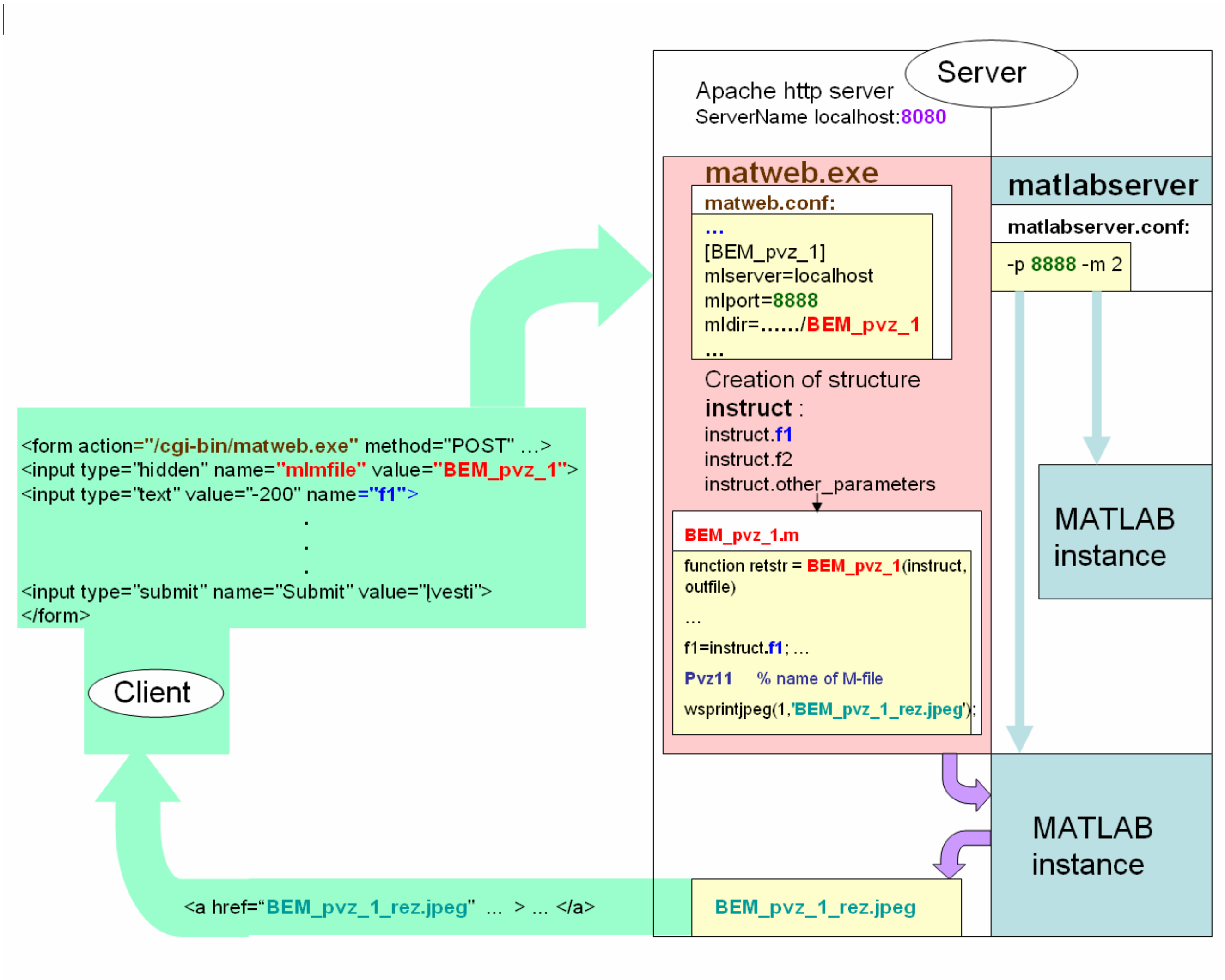


Figure 1. The scheme of MATLAB webservice BEM_pvz_1("Run FEA example in MATLAB")

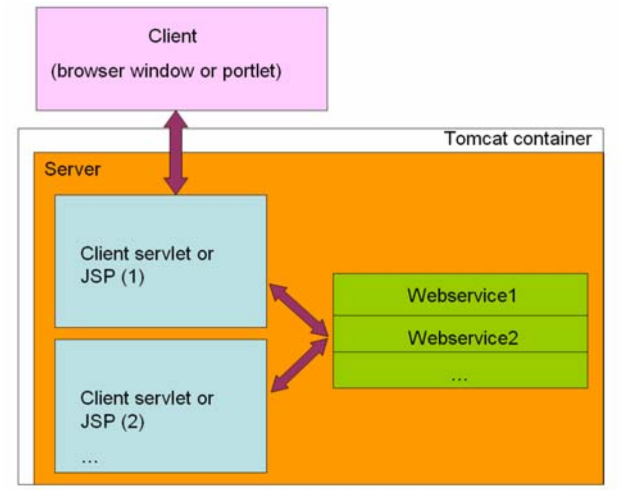


Figure 2. General client and server communication scheme by using JAXWS

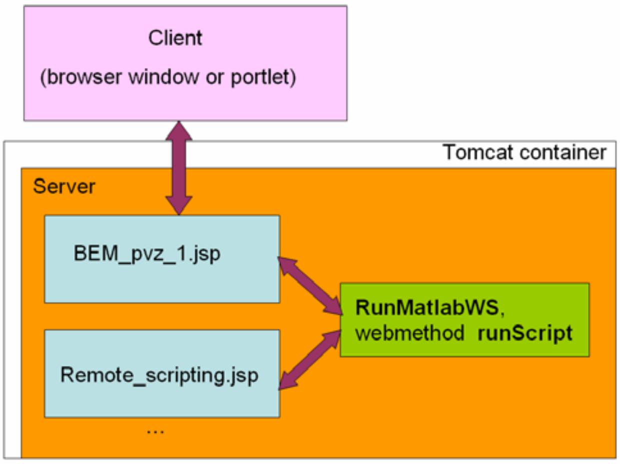


Figure 3. Client and server communication scheme in the case of FEA learning and research JAXWS web application

The functionality of Java-XML web services (WS) are created by means of service and client Java programs (classes). The client program may be realized as a servlet or Java Server Page (JSP), which can be directly activated via **`<action=JSPname>`** from HTML forms available in the client portlet or browser window. Client servlets may invoke web methods (WM) described in the WS.

The hearth of our WA once again is the remote invocation of **`MATLAB`**, which is able to accept and execute user's M-files sent over Internet. The **`MATLAB`** package does not contain the means of invoking the execution of **`MATLAB`** from Java, however, calls from `MATLAB` to Java are possible. We applied **`MatlabControl`**, **`MatServer`** and **`MatClient`** Java classes created by Kamin and Hui [10,11]. By means of the class methods the instance of **`MATLAB`** is provided, which resides continuously on the server and is able to accept and execute **`MATLAB`** M-files. Log file created during **`MATLAB`** execution contains the information, which is usually being output to the **`MATLAB`** command window.

As servlets, JSP, and WS are fully functional Java programs, the decision which functions should be delegated to WS and which are convenient to implement directly in the client servlet or JSP depends only on the designer of the WA. In our WA we are using one WS containing one web method (WM) **`runScript`**, which is based on **`createJob`** and **`finishJob`** methods available from **`MatClient`** Java class. All other functionality ensuring proper execution of the FEA example as a WA has been coded into the client JSP Figure 3.

Usage of JSP rather than the servlet as the client program is preferable because of large amount of static information contained in the client window, and certain fields, which are to be dynamically updated as the client-server communication goes on. Moreover, the JSP are directly reusable for creation of portlets in **`StringBeans`** portlet container used for deployment of GRID applications.

3.3 Functional characteristics of implemented web applications

The simplest problem of FEA learning course requiring the usage of WA is the analysis of results of a physically based model in accordance with the presented **MATLAB** script. Client(student) window contains references to learning materials and texts of sample scripts, input fields for entering the values of the parameters of the model and activation button for initiating **MATLAB** calculations on the remote computer(server). The view of the client portlet is presented in Figure 4. WA ***BEM_pvz_1x*** presents graphical and text results of calculations in separate windows by means of HTML references, and contains a field for presenting the log file of the execution process of the script. The client may clear the results files from the server. This functionality is achieved by creating two different activation directives in the same JSP page.

WA ***Remote_scripting***, which enables to run a **MATLAB** script created by the client on the remote computer(server) facilitates the solution of very different problems. The functionality is implemented by properly deploying the script files in **MATLAB** working directories. The WA performs the following tasks:

- Registers the user in the server and creates its working directory;
- Writes the scripts created by the user into the directory appointed for the particular user;
- Reads the indicated file from he server and places it into text area of the portlet for editing and modifying;
- Enables he user to clear indicated files from its work directory;
- Initiates the solution process;
- Presents the **MATLAB** log file for the user information and enables to view the text and graphical results files via appropriate HTML references.

Extension of possibilities of the WA enables to create other WA for initiation of other FEA programs (ANSYS, LSDYNA, COMSOL) on remote computers.

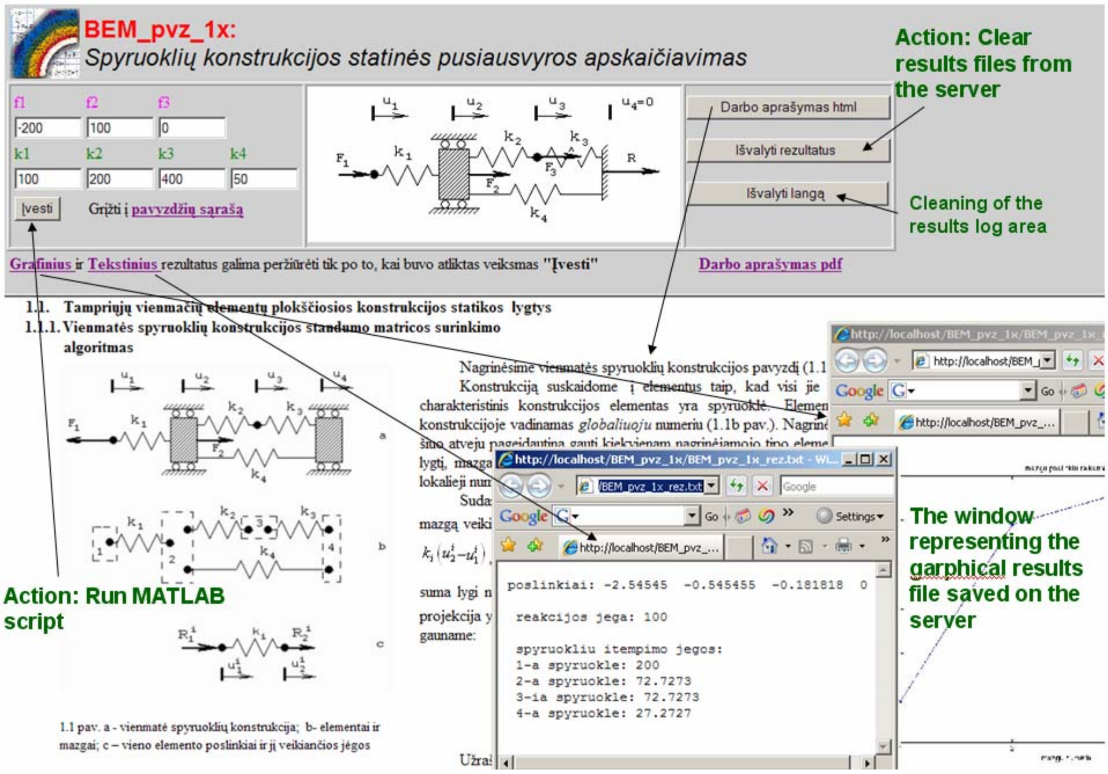


Figure 4. The view of the client portlet of the web application BEM_pvz_1x("Run FEA example in MATLAB")

4 Conclusion

The work presents the development results of finite element analysis distant learning and research tools implemented as web applications. Four different application cases comprising the scope of the activity of the virtual organization have been identified. Two example web applications have been implemented by using different web technologies – via the Common Gateway Interface by employing available service programs m*atlabserver* and *matweb* and by using Java-XML web services. The extension of the functionality of created web applications leads to creation of possibilities for remote execution of other FEA programs, such as ANSYS, LSDYNA, COMSOL. Among the important expected advantages of the created web applications the convenient and legal access to the licensed software for the users studying the FEA course and performing the research in the field of FEA should be mentioned. Simultaneously the created applications ensure distance access to e-textbooks and samples libraries.

References

[1] **Barauskas R**. Education in finite element analysis and applications in computational grid networks, *Advanced Learning Technologies and Applications – ALTA'03: Proceedings of the International conference, Kaunas University of Technology, September 11-12*, 2003, pp.34-38.

[2] **Banaś K., Płażek J.** Concepts for implementing adaptive finite element codes for grid computing, *Cracow Grid Workshop, 5-6 November*, 2001.

[3] Report "Activity model of virtual organization "Learning tools of FEA in GRID environment"", In: *Intermediate Activity Report "Creation, development and implementation of virtual science and studies organization on the base of GRID technologies",* Agreement Nr. ESF/2004/2.5.0-03/411/BPD-182/F450 BPD-182.

[4] ANSYS Basic Analysis Guide, Release 11 *Documentation for ANSYS*, ANSYS Inc. Southpointe, 2006.

[5] **LS-DYNA Keyword User's Manual, ver.970, Livermore Software Technology Corporation, 2003.**

[6] COMSOL Multiphysics User's guide, ver.3.3, COMSOL AB, 2007.

[7] **Foster I., Kesselman C., Tuecke S**. The Anatomy of the Grid. Enabling Scalable Virtual Organizations, http://www-106.ibm.com/developerworks/grid/library/gr-fly.html#IDA5BGQB

[8] **Sotomayor B.** The Globus Toolkit 4 Programmer's Tutorial, University of Chicago, Department of Computer Science ,2005.

[9] **Jendrock E., Ball J., Carson D., Evans I., Fordin S., Haase K.** The Java™ EE 5 Tutorial, Third Edition, For Sun Java System Application Server Platform Edition 9, Adison-Wesley, 2006.

[10] http://www.cs.virginia.edu/~whitehouse/matlab/JavaMatlab.html

[11] http://www.cs.utoronto.ca/~bowen/code/code.html