Journal of Universal Computer Science, vol. 18, no. 10 (2012), 1259-1278
submitted: 1/11/11, accepted: 23/5/12, appeared: 28/5/12 © J.UCS

Web-based Environment for Learning Discrete Event Simulation

Marijana Despotović-Zrakić, Dušan Barać
(Faculty of Organizational Sciences, University of Belgrade, Serbia
{maja, dusan}@elab.rs)

Zorica Bogdanović
(Faculty of Organizational Sciences, University of Belgrade, Serbia
zorica@elab.rs)

Branislav Jovanić
(Institute of Physics, University of Belgrade, Serbia
brana@ipb.ac.rs)

Božidar Radenković
(Faculty of Organizational Sciences, University of Belgrade, Serbia
boza@elab.rs)

Abstract: This paper describes a web-based environment for learning discrete simulation. The main goal of the paper is to foster the process of e-learning simulation by providing students and teachers with effective and comprehensive tools for creating, storing and executing discrete system simulation models. For these purposes the FONWEBGPSS application was developed and integrated into the e-learning system Moodle. The integration is implemented on three levels: users, processes and learning resources. The integration of users and processes is performed by synchronizing data in both systems. The integration of learning resources is performed by adjusting and implementing the IEEE LOM profile for learning simulation. In order to evaluate the impact of the proposed solution on the learning simulation outcome, research has been performed within the undergraduate course Simulation and simulation languages at the University of Belgrade. Research results indicate that students achieve better results in learning simulation when using FONWEBGPSS application integrated into the e-learning system than learning in a traditional way.

Keywords: Web-based simulation, learning simulation, simulation model web repository, discrete event simulation, e-learning system, Moodle
Categories: K.3.1, L.3, K.3.2

1 Introduction

Computer simulation has an important role in the modelling of real-world objects and processes. Important characteristics of a system, such as: stability, feasibility, performance, behaviour, etc. can be tested and analysed using simulation tools [Granlund et al., 2000]. In this way, it is easier to understand and predict functioning of real systems [Fonseca et al., 2009]. Nowadays, computer simulation plays a key role in the study of complex real systems, and therefore is included in most

engineering curricula [Kincaid and Westerlund, 2009]. Hence the need for effective simulation tools that ensure a simple and prompt creation of simulation models [Radenkovic and Markovic, 1999].

The main purpose of a software system for learning simulation is to enable cheap and easy learning and practicing environment, particularly suited for students of engineering. However, the majority of simulation tools have limitations such as platform dependence, high costs, maintenance difficulties, and limited reusability [Kuljis and Paul, 2000].

Ubiquitous Internet technologies provide numerous possibilities for improvement within the area of modelling and simulation. A Web-based simulation enables the simulation to be carried out over the Internet using a standard web browser and other resources and technologies offered by the Internet. [Byrne et al., 2010][Kuljisand Paul, 2003]. A Web-based simulation is defined as the use of resources and technologies offered by the Internet for ensuring an interaction with the client and server modelling and simulation tools. The main advantages of the web-based simulation approach are the following: efficient maintenance, increased integration, wide availability, interactivity, controlled access, collaboration, cross platform compatibility, and interoperability [Levytskyy et al., 2009] [Atri et al., 2008]. During the last decade, there has been much research related to developing tools, languages and methods for web based simulation [Fonseca et al., 2009] [Rossetti et al., 2009] [Tao et al., 2006].

Computer simulation has been studied for many years at the Faculty of Organizational Sciences, University of Belgrade within the scope of the Simulation and simulation languages course of study. The concept of e-learning can be successfully applied to learning simulations [Gaffney C., et al. 2010]. The crucial idea of this paper is based on developing a new web-based environment for learning discrete event simulation and integrating it in the system of e-learning based on the Moodle learning management system (hereinafter Moodle LMS).

In section 2, main aspects and considerations in teaching discrete event simulation in e-learning environment have been described. Section 3 contains details on the design and implementation of the FONWEBGPSS application, as well as on the integration with Moodle learning management system. Results of the evaluation of the developed environment for e-learning simulation are given in Section 4. Finally, the summarized benefits of the developed application, as well as future research directions are given in Conclusion.

2 Teaching discrete event simulation

A discrete system is one in which the state variable change only at a discrete set of points in time, while in a continuous system the state variables change continuously over time [Banks et al., 2010]. Complex systems such as: supply chain management systems, supermarkets, hospitals, airports, banks, factories, etc. where changes of the system's state occur at discrete points of time, are typical examples of discrete event systems. Discrete simulation provides engineering students with an opportunity to test and investigate various types of systems, look into the structure and interactions among the system components, analyze the design and implementation of the system, verify different solutions and environments, test system performances, etc. These

tasks are also important for students of information systems and technologies because they can learn to design many complex information systems and applications that include a variety of requirements. Acquired knowledge from modelling and simulation of real-life problems improves students' results in software solution development.

2.1 Software and languages for discrete event simulation

The major issue in the realization of courses that deal with discrete event simulation is to find the appropriate language and tools for simulation.

The GPSS (General Purpose Simulation System) is a language interpreter for executing the simulation of discrete, stochastic systems [Stahl, 2001]. In the GPSS, the language model is created by language commands [Charalambides, 1984]. Each program in the GPSS is described through processes, sequences of activities and operations that should be executed on the different attributes of the model. Widely known versions of the GPSS language for personal computers are: GPSS World [Cox, 1991]. GPSS/PC [Chisman, 1993], GPSS/H [Abed et al., 1985].

GPSS/FON [Radenkovic and Markovic, 1999] is an implementation of the GPSS language created in the Laboratory for simulation, at the Faculty of Organizational Sciences, University of Belgrade. Due to its simplicity, a quick and easy model debugging, the GPSS/FON language has been successfully implemented in teaching the Simulation and simulation languages course for many years. One of the main reasons for implementing our own version of the GPSS simulator was related to cost of licenses for commercial GPSS implementations. Licenses would have to be provided for more than 100 computers at the Faculty of Organizational Sciences and for more than 300 students each year. The GPSS/FON application is free and both installation and source code can be downloaded from our website. The GPSS/FON application is also used at other Faculties within the University of Belgrade.

However, during the period of exploitation, we have noticed some of GPSS/FON disadvantages. The basic problem with GPSS/FON is that it is a desktop application, therefore it needs to be compatible with the operating system of students' computers. With each new version of the operating system it is possible that a new version of GPSS/FON be required. Another disadvantage of the GPSS/FON is that as a desktop application it can hardly be integrated with the e-learning system Moodle.

In order to overcome these disadvantages, we have developed a new application, named FONWEBGPSS. The architecture and implementation of FONWEBGPSS application are described in Section 3.

2.2 Course description

The FONWEBGPSS application is used for teaching discrete event simulation in the scope of the Simulation and simulation languages study course. The course is organized at the fourth year of undergraduate studies in the area of information systems and Internet. Before attending this course, students are obliged to take several exams in the field of programming and are familiar with programming concepts and several programming languages (Java, C). The course deals with the basic concepts and applications of computer simulation.

The course is realized using the combination of traditional classroom-based teaching and e-learning technologies, i.e. blended learning [Dondi C, 2009]. The Moodle learning management system has been used for synchronizing teaching activities and managing online courses. The Moodle LMS provides teachers and students with communication and collaboration framework that supports features, such as: e-mail, messaging, forum, notice boards, calendars, evaluation reviews. This can increase the interaction among students and between students and instructors. Learning resources are presented through tutorials, multimedia presentations, online lessons, examples, etc.

The course curriculum contains three main areas: discrete event simulation, continuous simulation, and 3D simulation. Students create their own models and solutions for real-life problems related to the areas of business, industry, economy, etc. Students' work during the course is evaluated through five projects, which are scheduled in the course of the semester. Three projects cover the area of discrete event simulation, while the forth project is within the area of continuous simulation. The last one deals with 3D modelling. Using appropriate simulation tools in solving practical problems motivates students and contributes to overcoming possible problems with mathematical difficulties.

2.3 SCORM compliant repository for learning simulation

Recently, there have been a lot of reasearch in the area of web-based learning related to establishing standards and creating learning content in a common way. The Sharable Content Object Reference Model brings together the existing e-learning standards in a single distributable package. The SCORM provides a structured design pattern related to the creation of interchangeable educational material. The SCORM supports integration of learning resources with the learning management system [Chang et al., 2004][Zhu, 2007].

The creation of well structured descriptions of learning resources is a critical issue in developing e-learning systems. These descriptions should enhance creation, retrieval, delivery, and reusing of learning resources. Metadata provide a common nomenclature for learning resources to communicate and exchange with others in a common way. A good metadata model needs completeness, carefulness and flexibility [Chang et al., 2004]. The most used metadata model is the IEEE LTSC LOM. It defines nine categories to describe learning resources [Learning Technology Standards Committee of the IEEE, 2002]. The categories include: General, Classification, Annotation, Lifecycle, Technical, Meta Metadata, Educational, Relation and Rights. According to the IEEE, a learning object is "any entity, digital or non-digital, that may be used for learning, education or training". The LOM data model defines which attributes of a learning object should be described and what vocabulary should be used in these descriptions.

Every course that teaches simulation includes a variety of simulation models. This study aims to create a simulation model repository, where all the models should be available via web. Repositories enable users to store, manage and efficiently access learning resources [Namuth et al., 2005]. However, the existing standards do not provide enough attributes to treat simulation models as learning objects. This makes their integration into learning resources repository a hard task. Each learning object repository covers a particular area of interest. Accordingly, when creating learning

objects, teachers should take into consideration which group of attributes, i.e., metadata elements and value sets is appropriate to be included in the learning objects. The specification of these metadata elements and value sets is called an "application profile". Profiles are used to adapt metadata specifications and vocabularies to the requirements of the local community [Najjar et al., 2003]. The creation of an application profile allows the community of users to specify which elements and vocabularies they will use.

In this paper we address this issue, that is, we propose an IEEE LOM Computer Simulation Learning Application Profile that can be used for tagging computer simulation learning resources. We have identified controlled vocabularies that can indicate possible extensions to the IEEE LOM Standard concerning science curriculum properties. Further, we created an online repository of computer simulation examples.

There are three permitted approaches to extend the LOM within a SCORM manifest file [Mason, 2006]: (a) new metadata elements, (b) new vocabulary values, and (c) the reference of an internal or external XML file using a location element. As proposed in literature, the best solution is to extend vocabulary values [Sampson and Zervas, 2008][CLEO Collaborative Partners, 2003]. The process of creating Computer Simulation Learning Application Profile is presented in Figure 1.

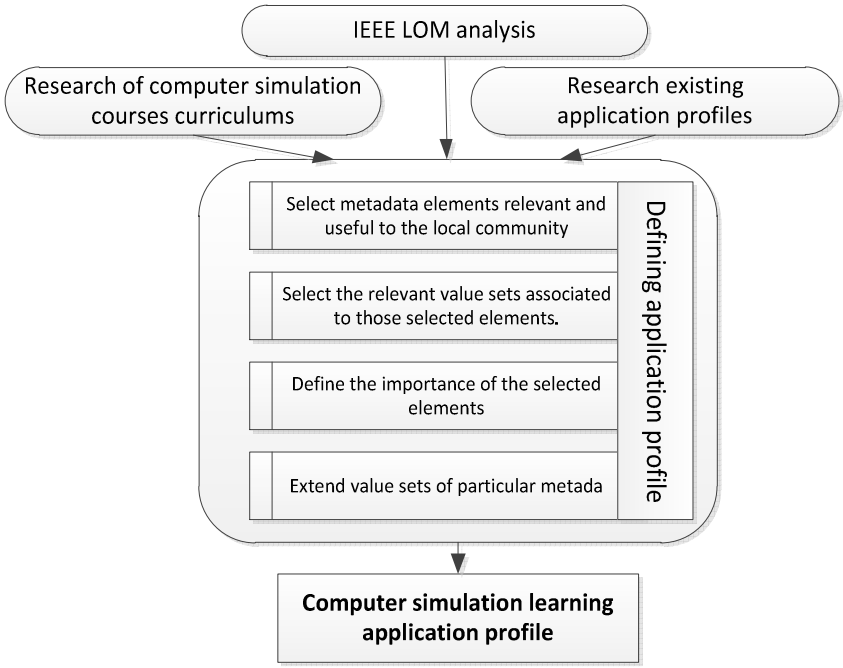

Figure 1: Creating computer simulation application profile

The first step includes a thorough analysis of IEEE LOM categories, attributes and vocabulary sets. The main areas of computer simulation studies are related to:

continuous simulation, discrete event simulation and 3D modelling. The existing application profiles extend the SCORM LOM mainly with respect to Classification category or by making some elements optional [Mason, 2006]][CLEO Collaborative Partners, 2003][Zhu, 2007]. These extensions are implemented through adding new elements into a particular category or by extending element vocabularies [Learning Technology Standards Committee of the IEEE, 2002]. A vocabulary is a recommended list of appropriate values for the element.

The created computer simulation application profile is presented in Figure 2. Four categories are defined as mandatory: General, Technical, Educational and Classification, while the others are optional. The IEEE LOM Classification category describes a learning object in relation to a particular classification system. Our application profile adds controlled vocabularies *Type of simulation* and *Application area* to sub-element *9.1 Purpose*. The *Type of simulation* vocabulary has three different states: continuous simulation, discrete event simulation and 3D modelling. For each of the states, a further explanation through *problem type* value set is given. For instance, discrete event simulation models studied in the scope of the Simulation and simulation languages course can be grouped into three types: one serving place and one process, two or more serving places and one process, and parallel processes. The *Application area* defines business or scientific area where simulation model could be applied (industry, information system, market, etc.)

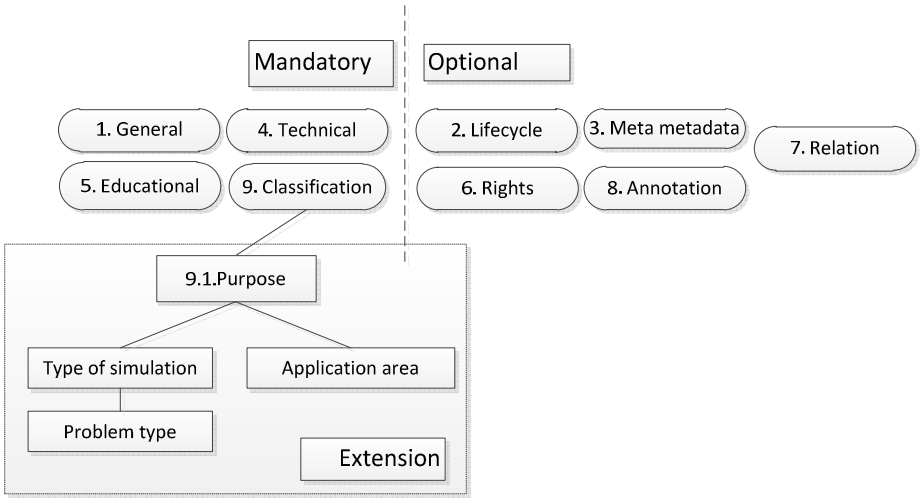

Figure 2: Computer simulation application profile

Each example of the simulation is presented as SCORM PIF [Chang et al., 2004]. The manifest itself is an independent XML file that must be named *imsmanifest.xml*. It describes four parts of the learning object: metadata, organizations, resources and sub-manifests. As recommended by the IMS CP, the manifest should be stored at the directory top level, from where other sub-folders are created and organized along with the physical files of the learning unit (e.g. PDF, HTML, JPG, MPEG, etc.). Figure 3 shows the SCORM compliant package for one of the discrete simulation models in

the implemented repository. The model is related to computer network simulation. The manifest file based on described application profile is presented in the right part of the Figure 3.

```
<manifest identifier="com.scorm.manifesttemplates.scorm2004.4thEd" version="1"
          xmlns = "http://www.imsglobal.org/xsd/imscp_v1p1"
          xmlns:adlcp = "http://www.adlnet.org/xsd/adlcp_v1p3"
          xmlns:lom="http://ltsc.ieee.org/xsd/LOM"
          xsi:schemaLocation = "http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v
                                http://www.adlnet.org/xsd/adlcp_v1p3 adlcp_v1p3.
                                http://www.adlnet.org/xsd/adlseq_v1p3 adlseq_v1p
                                http://www.adlnet.org/xsd/adlnav_v1p3 adlnav_v1p
                                http://www.imsglobal.org/xsd/imsss imsss_v1p0.xs
                                http://ltsc.ieee.org/xsd/LOM lom.xsd" >
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>1.2</schemaversion>
    <lom:lom>
      <lom:general>
      <lom:technical>
      <lom:educational>
      <lom:classification>
        <lom:purpose>
          <lom:TypeofSimulation language="en-US">Descrete event simulation</lom:Typeofs
          <lom:ApplicationArea language="en-US">Computer networks</lom:ApplicationArea>
        </lom:purpose>
      </lom:classification>
    </lom:lom>
  </metadata>
  <organizations default="Descrete simulation">
  <resources>
</manifest>
```

Figure 3: Example of simulation model as SCORM package

All simulation models are stored as SCORM packages in our web repository. Each simulation model is a zip file that can be accessed via the web application or integrated in the Moodle LMS.

3 Design and implementation

The main component of the web environment for learning and teaching simulation is the FONWEBGPSS. This web application provides a user-friendly environment for modelling and simulation in the area of discrete event systems. Using the application, students are able to create and adjust simulation models, execute simulation and perform a simulation results analysis in a web page. Particular features of the web application include: improved performance of the teaching activities, integration of all the activities in the teaching process, studying simulation by a variety of real models, a simple way of learning simulation that is accessible to all students at any moment, an interactive environment with a resident editor, processor and result analyzer, fast and easy model debugging, different views and analysis of simulation results.

3.1 Application development process

The process of FONWEBGPSS web application development is shown in Figure 4. After analyzing the requirements and designing the architecture of the system, each component of the system has been implemented and tested. The final phase in web application development was the integration with the Moodle LMS.

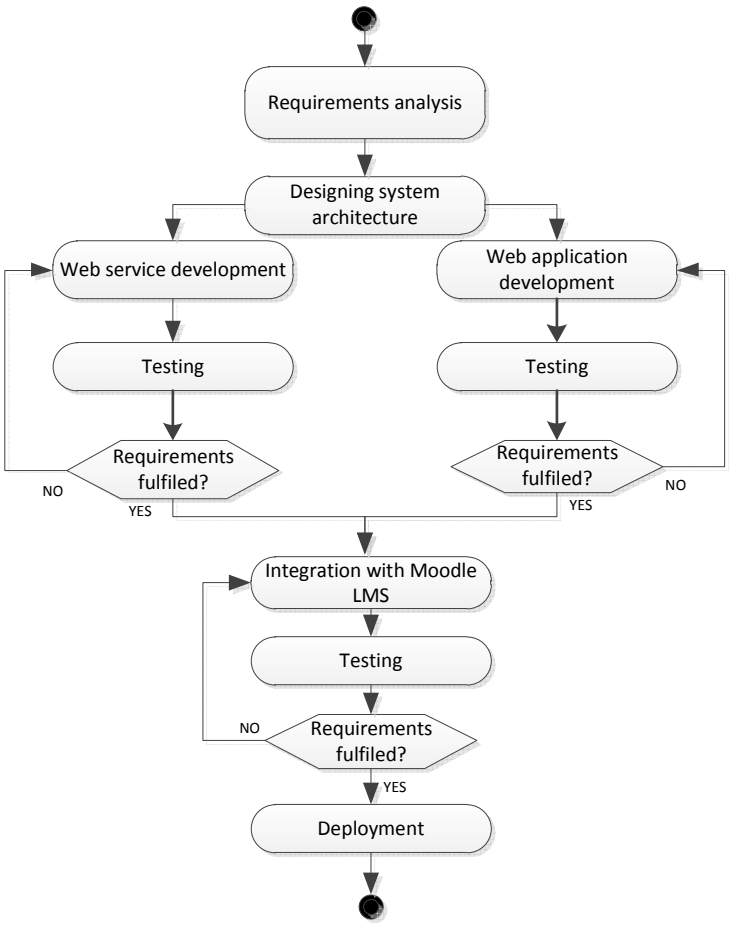

Figure 4: FONWEBGPSS development process

The most important requirement in the FONWEBGPSS application design was that all functionalities from the desktop version of the software GPSS/FON had to be kept. The goal was to deliver the existing functionalities through a new web interface. The new application had to use the same simulator code as the old one [Radenkovic and Markovic, 1999].

3.2 Conceptual model of the application architecture

The architecture of the proposed environment for learning simulation is presented in Figure 5. The FONWEBGPSS application was developed with respect to modern principles of software development.

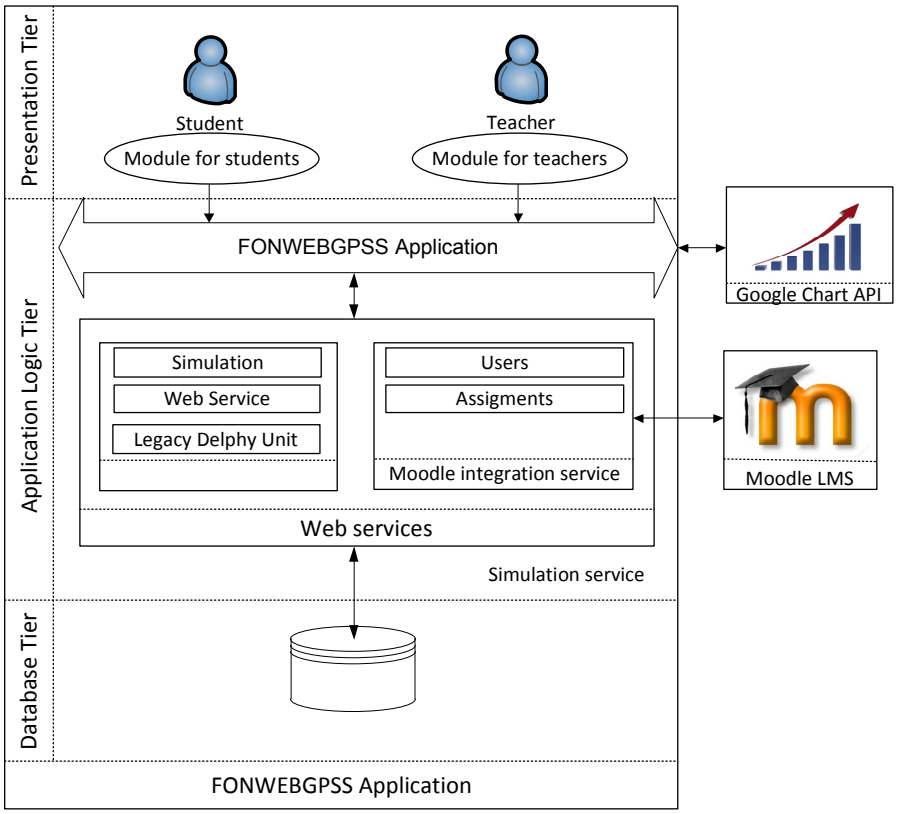

Figure 5: Architecture of the web based environment for learning simulation

The main components of the system are: the FONWEBGPSS application, the Moodle LMS, the web service for performing simulation, and the web service for integration with the Moodle LMS. The users' requests are created on a web page and submitted to the application logic tier. The key part of the application is the component marked as FONWEBGPSS (Figure 5). This component implements all the features of the application. It is responsible for the following tasks: receiving users' requests, communication with the database, managing application features, invoking web services that perform simulation, receiving simulation results, sending data to the component that draws simulation statistics. Dynamic web technologies (ASP.NET, ADO.NET, XML) were used in the implementation of application logic tier. The code is written in C#. The interface toward the FONWEBGPSS Web service component, Google Chart API wrappers, the DataSet access to data base and the class for generating diagrams are implemented through additional classes.

The web service for simulation is a Delphi2007 project that consists of two components: web service and GPSS library. The web service tier receives the data sent from the application logic tier and calls the method that executes simulation. One

of the units in the web service contains the implementation of the web method that starts the simulation and forwards the results to the application logic layer in the XML format. The GPSS library contains an original GPSS code. The original GPSS files were transformed from the Pascal to the Delphi language files. The FONWEBGPSS simulator is derived from an older version of the GPSS/FON simulator and integrated into the web service. The Web service implements a three-phase simulation [Radenkovic and Markovic, 1999].

3.4 Integration with Moodle LMS

The integration between the FONWEBGPSS and the Moodle LMS is implemented with respect to three aspects: user accounts, processes and learning resources (Figure 6).

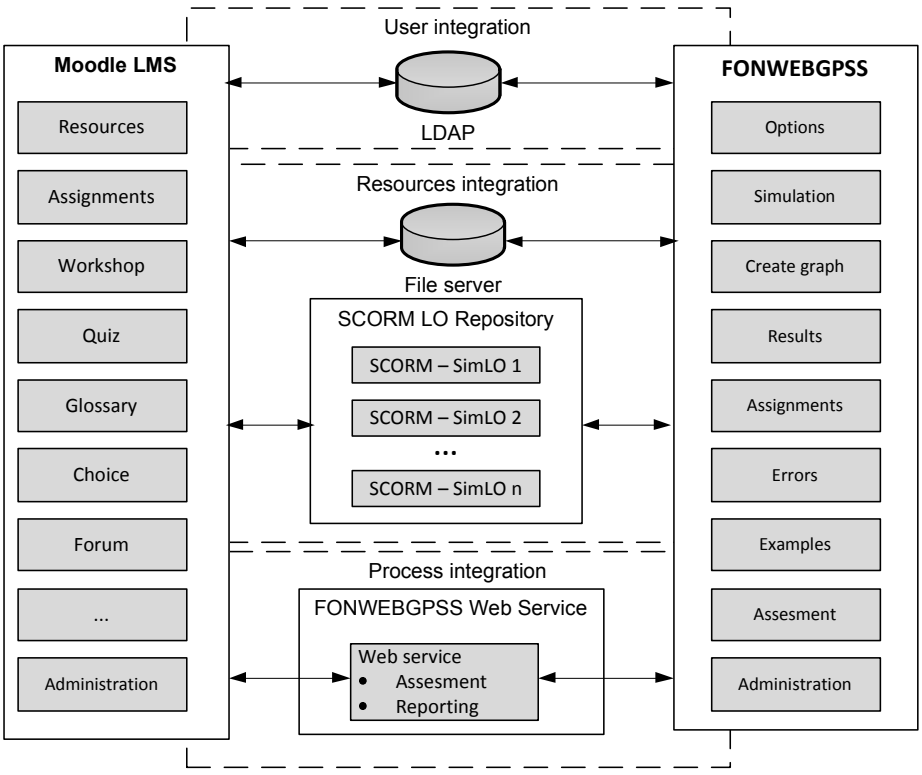

Figure 6: Integration of FONWEBGPSS application and Moodle LMS

The integration of user accounts is based on the LDAP (Lightweight Directory Access Protocol). The FONWEBGPSS application and Moodle LMS use the same resource for managing user profiles and a single-sign on is implemented. When a user logs into Moodle, they are automatically logged into the FONWEBGPSS, and vice-versa. The user roles are synchronized in both applications. In this way, if the user

gets a role (student, administrator, and teacher) in the Moodle, they get the same role in the FONWEBGPSS.

The process integration is reflected in the coordination of learning activities within the FONWEBGPSS and the Moodle LMS. The application is placed on the main page of the Moodle course, Simulation and simulation languages, as an additional module (Figure 7.). This module is available for all students that attend the courses. By clicking on the FONWEBGPSS link in the upper right corner of the page, students and teachers are redirected to the start page of the FONWEBGPSS application.

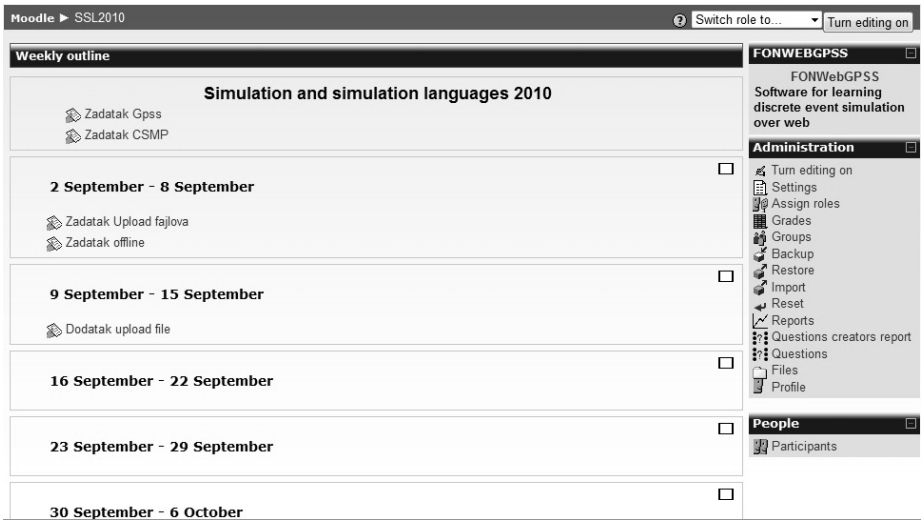

Figure 7: Homepage of the Simulation and simulation language course with FONWEBGPSS module

When a student submits their homework within the application, it can be seen in the Moodle LMS, too. Teachers can choose whether they want to assess students' assignments in the Moodle course or directly in the FONWEBGPSS. The process related to assignments upload and assessment is presented in Figure 8. The diagram shows the sequence of actions and the data synchronisation.

All data about students' results within the application are available via the Moodle LMS. This enables teachers to perform different statistic analyses related to students' activities.

Considering the resource integration, the learning object repository has been created according to the adjusted SCORM model described in section 2.1. The FONWEBGPSS application provides students with a lot of examples of simulation models. All the models are implemented as SCORM PIF and stored in the learning object repository. These files are SCORM compliant, so they can be easily integrated in the Moodle LMS. Further, when students upload their assignments within the FONWEBGPSS application, a text file that contains the programme code is automatically created. The file is stored in the appropriate Moodle LMS directory

where all the other files related to students' work and assignments in the Moodle LMS are commonly stored.

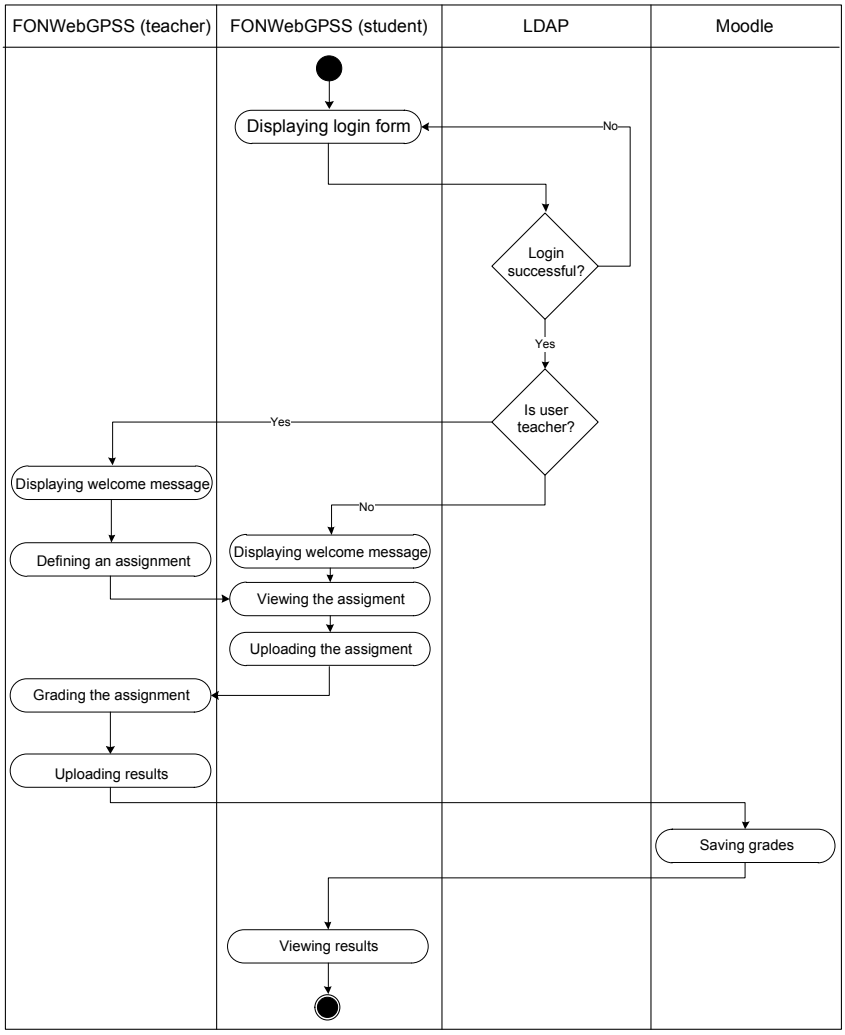

Figure 8: Integration of Moodle LMS and FONWEBGPSS application within process related to assignments upload and assessment

3.5 Platform

The FONWEBGPSS application is used by more than 200 students each semester. Accordingly, it requires powerful hardware and software resources, particularly two web services for integration with the Moodle LMS and simulation execution. The FONWEBGPSS application is hosted on Windows Server 2008; it runs on IIS 6.0 web server, while the database is created in SQL Server 2005. On the other hand,

Moodle is hosted on the Linux platform, the Apache web server and uses a MySQL database.

The FONWEBGPSS application and two web services are hosted on the cloud computing infrastructure of the Laboratory for Simulation. Cloud computing is an area of computing where IT scalable capacity is provided in the form of services delivered via the Internet. A private cloud has been developed as infrastructure for our e-learning system. The private cloud is deployed using the following: Centos 6.1 64bit – Operating system, Xen – hypervisor, OpenNebula – virtual infrastructure management system, MooseFS – distributed file system, Ganglia – automated monitoring system, Haizea – an open source lease management architecture which enables the provision of resources required for teaching and laboratory exercises. The private cloud consists of the server that hosts all described software tools and physical computers (nodes) and their recourses.

3.6 User interface

The FONWEBGPSS features are grouped into two parts: a module for teachers and a module for students. The course teacher can create and test simulation models, assess students' works, and perform different types of statistical analysis. Further, teachers can also do common system administration tasks such as: managing grades, user accounts, etc. These options are placed in the main menu (upper part of the window).

The part of the module for teachers with options for managing resources within the simulation learning object repository is shown in Figure 9. There is a list of existing examples in the repository. The teacher can add, edit or remove examples. Furthermore, teachers can view students' submissions, perform simulation and check results by a simple click on the student's simulation model. The teachers' work is more efficient in comparison with the old desktop application, because all actions can be performed in only a few clicks.

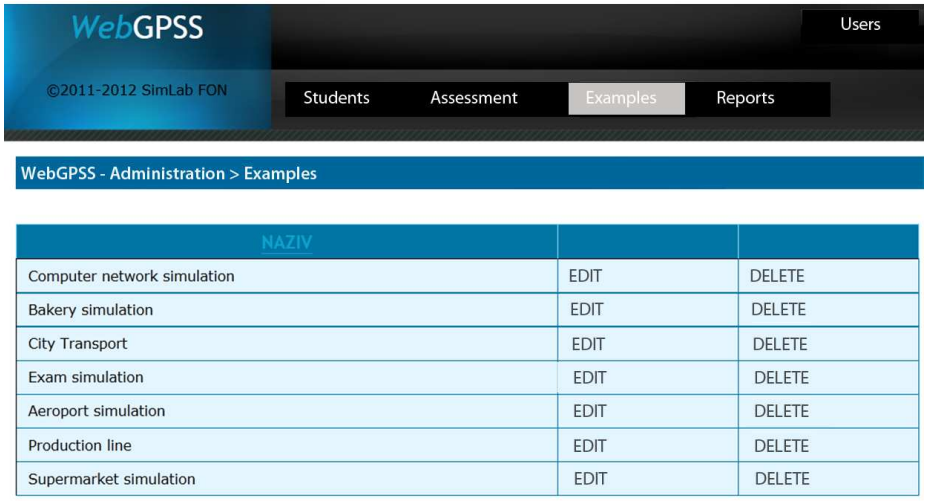

Figure 9: Module for teachers

Provided within the module for students are options for creating and testing simulation models, performing simulation, configuring graphical environment, and analysing simulation results. When students finish simulation projects, they can submit their works via the application and view grades later. After logging into the application, the student creates a verbal model in the Description Tab of the web page shown in Figure 10.

Figure 10 also shows the icon for opening source code editor, located in the upper left corner of the page, and the main menu of the application organized respectively in tabs: Source code editor, Results, Errors, Tables, Graphs, Description.

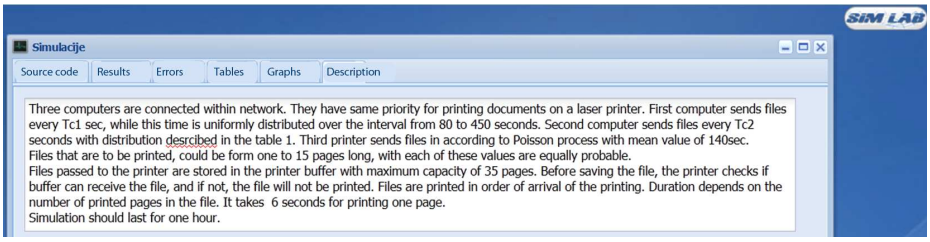

Figure 10: Module for students – Description tab

A student creates a GPSS model by typing GPSS commands in the source code editor (Figure 11). The central part of the window is reserved for typing the source code using GPSS commands and comments. The code editor is enriched with an AJAX autosuggest feature. When the student starts to type a command in the editor, the application itself suggests available commands related to the typed letters. Under the source code editor, there is a drop-down list where student can choose from the predefined types of problems, and below this option, there is a textbox where the title of the problem should be typed.

After the simulation model is created and all the required data are provided, the simulation can be executed by selecting the option (button) *simulate*.

Finally, the student gets a notice about whether the simulation succeeded or failed. If there are any syntax errors, the student will get an error message. If there are no syntax errors, results are displayed in the Results tab (Figure 11). By default, results are presented in the text format. However, students are enabled to view the simulation results in the form of tables or charts. The Google Chart API technology was used for drawing charts and statistics within the web page.

A chart can be created for every storage, queue, table or other GPSS concept used in the model (Figure 12). Students can select the type of the chart (line chart or histogram) and the data that will be shown on the chart. Various variables of the simulation model can be presented, such as: average utilization, average contents, number of entries and average time per transaction for each storage and queue in the model. The data value can be presented through frequencies, percents and cumulative percents, for each table.

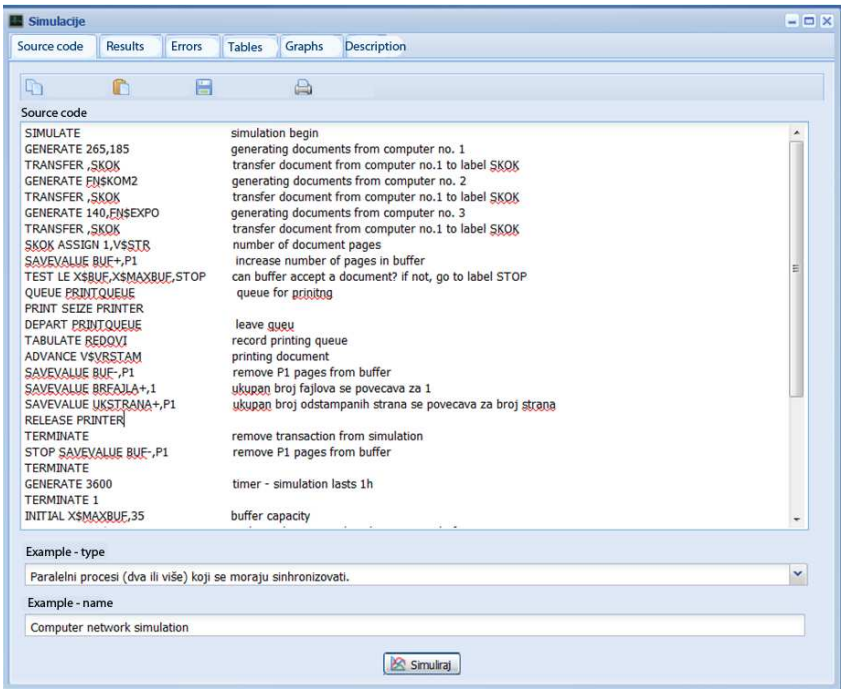

Figure 11: Module for students - Source code editor

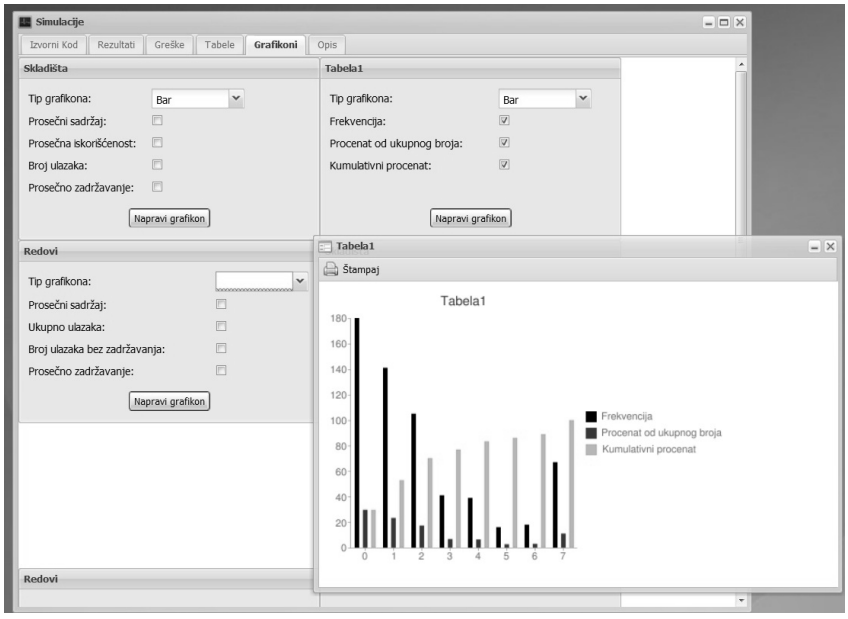

Figure 12: Module for students - Results in the form of graphs

4 Evaluation

This study aims to investigate the extent to which the implementation of the FONWEBGPSS application impacts the students' results on knowledge tests in the area of discrete simulation. The experiment was conducted during the spring semester in the school year 2011/12 on a sample of 60 undergraduate students of the Faculty of Organizational Sciences, University of Belgrade. Students who attended the Simulation and simulation languages course were randomly divided into experimental (30 students) and control groups (30 students). The sample included 40% female and 60% male students. All students, both experimental and control groups had a block of lectures related to the course topics presented in the traditional way. Then, the students from the experimental group attended labs through the e-learning system and the FONWEBGPSS application integrated in the system, while the students from the control group attended labs using the old GPSS/FON application. Finally, students were given a knowledge test.

In order to measure the research results, we have used the knowledge test that students take at the end of semester and a questionnaire for students. In the knowledge test, students solve problems from the area of discrete systems modelling and simulation. A typical task in the test includes the following requirements: creating simulation model, verbal description of created model, and, editing simulation model and adjusting it to specific requirements. The test we applied in the experiment is a standard test used for testing students for more than five years. The students in the experimental group took the final test using the FONWEBGPSS applications, while the control group students took the final test in the standard form, i.e. on paper.

The students also completed a questionnaire after taking the exam. The aim of the questionnaire was to gather data about students' impressions of the FONWEBGPSS application. We wanted to assess the level of gained knowledge in the area of discrete system simulation, the FONWEBGPSS application quality and the impact of the FONWEBGPSS application on the overall learning process outcome.

A descriptive comparative statistics of results achieved on knowledge test is presented in Table 1 (N=30 for each group).

	Experimental group	**Control group**
Mean	8.97	8.43
Std. Deviation	0.928	1.406

Table 1: Mean and Std. Deviation for knowledge test

The results show that there is a larger number of students from experimental group that achieved high marks in comparison with the number of students from the control group. $F(2,58)=7.7036$ ($p<0.05$). The results also show that the following factors had an impact on the students' results: *average mark*, i.e. students that achieved good results in the past, got high marks on knowledge test in the area of discrete simulation ($F(2,58)= 4.498$ (sig.$=0.11<0.05$)); *the time student spends in learning during the semester,* i.e., students that learn more frequently achieved better results on test

(F(2,58)=2.824 sig.=0.048<0.05)); *the type of learning material student use in studying*. i.e. research results showed that students who learned using video presentations achieved best results on the test (F(2,58)= 1.682, (sig.=0.49<0.05)).

However, the analysis of results shows that the factors related to the frequency of using information and communication technologies (hereinafter ICT) (F(2,58)= 0.346, (sig.= 0.793>0.05)) and *LMS in learning* (F(2,58)= 0.393, (sig.= 0.759>0.05)) do not have an impact upon the results the students achieved. This result can be explained by the fact that quite a large number of students use the ICT and LMS in the learning process every day.

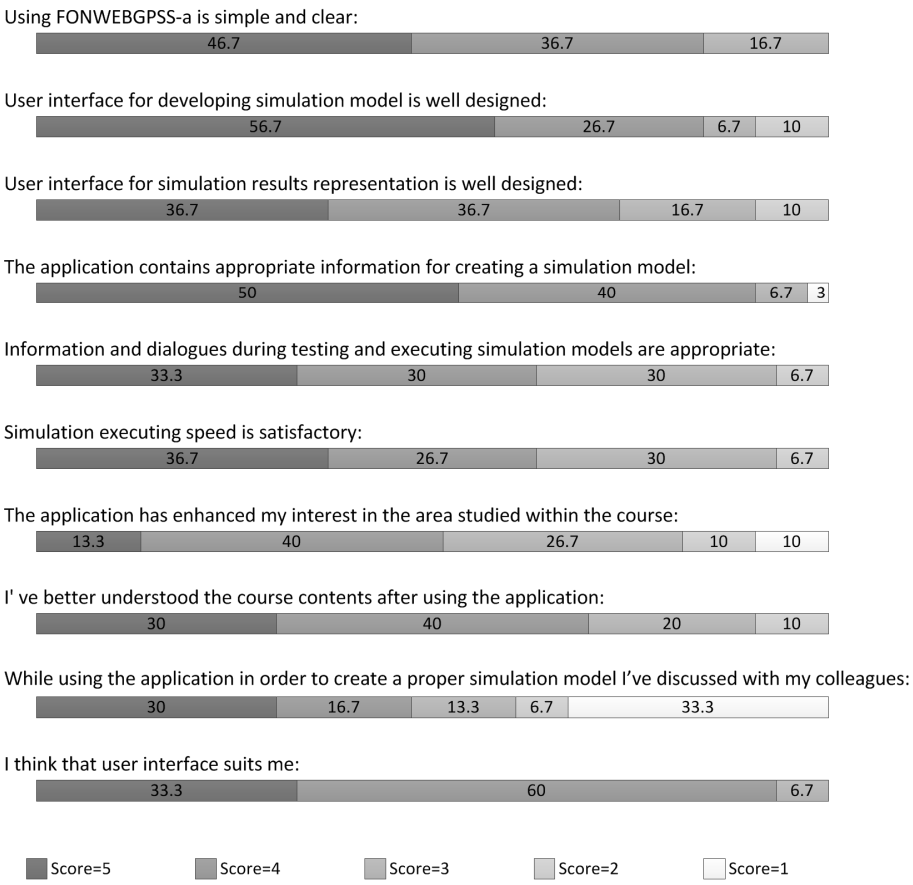

Figure 13: Results on the FONWEBGPSS application quality and learning experience(in %)

Further, parameters such as: *the number of examples studied before an exam* (F(2,58)= 0.396, (sig.= 0.757>0.05)), *attendance in class* (F(2,58)= 0.822, (sig.= 0.594>0.05)), and *the time spent in pre-exam preparation tasks* (F(2,58)=0.576, (sig.= 0.636>0.05)), did not have an impact on the students results. Thus the only factor related to the learning habits that had an impact on the students' achievement was the

time spent in learning during the semester. Better results on knowledge test in the area of discrete simulation were achieved by students that studied the learning materials on a more frequent basis. This conclusion suggests that students should be motivated to work more often through projects and other tasks during the semester.

The following variables were used to assess the FONWEBGPSS quality and impact on learning outcomes: the quality of user interface applications, the user interface design, the quality of the application features, the application speed, the students' interest in the area of study, their understanding of the course content, and the collaboration among students. Figure 13 shows the percentage of students' scores for each question (5-highest; 1-lowest). The total of 30 students filled the questionnaire. On the basis of the research results, a conclusion can be drawn that students are generally satisfied with the user interface. However, the speed of the simulation execution is not at the highest level, and the presentation of the simulation results can be improved. Students have found that the application does not encourage discussion and collaboration, so these aspects of the application should be improved.

5 Conclusion

Numerous studies emphasized the benefits of web-based simulation [Byrne et al., 2010]. There has been a significant amount of research related to the development of tools and applications that enable a simulation execution via the web. This study describes an approach to fostering the process of e-learning and teaching simulation. Students and teachers were provided with effective and comprehensive tools for creating, storing and executing discrete system simulation models via a web application. The main contribution of the study is reflected in the web-based environment for learning discrete event simulation that enabled a seamless integration of the FONWEBGPSS web application into the e-learning system. The integration of these two solutions was implemented on three levels: users, processes and learning resources. So far, not many attempts have been made to integrate a simulation tool into the e-learning process [Byrne et al., 2010]. Mainly, researches focus only on simulation performances and application features, but not on learning outcomes [Kincaid and Westerlund, 2009]. The evaluation results show that students had achieved better results in studying simulation when using the FONWEBGPSS application integrated into the e-learning system than when learning in the traditional way. Students rated the application interface and features very highly. Further, we proposed a tailored IEEE LOM profile for learning simulation. The Web repository of the SCORM compliant simulation models was provided within the application.

However, a few constraints related to using of the web application in the e-learning process have been noticed. The application did not have a positive impact on the level of students' collaboration. The provided web environment for learning simulation did not enhance students' interest in the area studied within the course. Accordingly, the future research is primarily directed toward the development of additional features for collaboration, as well as towards providing interactive graphical elements and introducing edutainment concepts that would enable students to learn in a more interesting way. Finally, the application performances should be improved by an overall use of cloud computing infrastructure.

Acknowledgment

The authors are thankful to Ministry of Education and Science of Republic of Serbia for the financial support, grant number 174031.

References

[Abed et al., 1985] Abed, S.Y., Barta T.A., mcRoberts K.L.: "A qualitative comparison of three simulation languages: GPSS/H, SLAM, SIMSCRIPT", Comput. Ind. Eng., 9, 1 (1985) 35-43.

[Atri et al., 2008] Atri, A., Breitenecker F., Nagele N., et al.: "Distributed discrete simulation on the web", Proc. of EMSS 2008, Amantea, Italy, (2008) 392-397.

[Banks et al., 2010] Banks, J., Carson J.S., Nelson B.L., Nicol D.M.: "Discrete-Event System Simulation 5th Ed"; Prentice Hall (2010)

[Byrne et al., 2010] Byrne, J., Heavey C., Byrne P.J.: "A review of Web-based simulation and supporting tools", Simul. Model. Pract. Th. 18, 3 (2010) 253-276.

[Chang et al., 2004] Chang, W.C., Hsu, H.H., Smith, T.K., Wang, C.C.: "Enhancing SCORM metadata for assessment authoring in e-Learning", J. Comput. Assist. Lear., 20, 4 (2004) 305-316.

[Charalambides, 1984] Charalambides L.C.: "The development of complex GPSS programs: An empirical assessment", Comput. Ind. Eng., 8, 2 (1984) 153-163.

[Chisman, 1993] Chisman, J.A.: "Introduction to simulation modeling using GPSS/PC", Computers & Operations Research, 20, 5 (1993) 557.

[CLEO Collaborative Partners, 2003] CLEO Collaborative Partners: "CLEO Extensions to the IEEE Learning Object Metadata", Version 1.1, CLEO White Paper CLEO001, (2003).

[Cox, 1991] Cox, S.W.: "GPSS World: a brief preview", Proc. of the 1991 Winter Simulation Conference, Phoenix, USA (1991) 59-61.

[Dondi, 2009] Dondi, C.: Innovation and Quality in e-Learning: a European Perspective, Journal of Universal Computer Science, 15, 7 (2009) 1427-1439.

[Fonseca et al., 2009] Fonseca, P., Juan A.A., Pla L.M. et al: "Simulation education in the internet age: some experiences on the use of pure online and blended learning models", Proc. of the 2009 Winter Simulation Conference, Austin, USA (2009) 299-309.

[Gaffney, et al. 2010] Gaffney, C., Dagger D., Wade V.: "Authoring and Delivering Personalised Simulations – an Innovative Approach to Adaptive eLearning for Soft Skills", Journal of Universal Computer Science, 16, 19 (2010), 2780-2800

[Granlund et al., 2000] Granlund, R., Berglund E., Eriksson H.: "Designing web-based simulation for learning", Future Gener. Comp. Sy., 17, 2 (2000) 171–185.

[Kincaid and Westerlund, 2009] Kincaid, J.P., and Westerlund K.K.: "Simulation in education and training", Proc. of the 2009 Winter Simulation Conference, Austin, USA (2009) 273-280.

[Kuljis and Paul, 2000] Kuljis, J., Paul R.J.: "A review of web-based simulation: whither we wander?" Proc. of the 32nd conference on Winter simulation, Orlando, USA (2000) 1872-1881,

[Kuljisand, 2003] Kuljis J., Paul R.J.: "Web-based discrete event simulation models: Current states and possible futures", Simulat. Gaming, 34, 1 (2003) 34-39.

[Learning Technology Standards Committee of the IEEE, 2002] Learning Technology Standards Committee of the IEEE: "Draft Standard for Learning Object Metadata", Institute of Electrical and Electronics Engineers, 2002.

[Levytskyy et al., 2009] Levytskyy, A., Vangheluwe, H., Rothkrantz, L.J.M., Koppelaar, H.: "MDE and customization of modeling and simulation web applications", Simul. Model. Pract. Th. 17, 2 (2009) 408-429.

[Mason, 2006] Mason, B.: "Digital Libraries in Support of Science Education: A Case for Computational Physics", Comput. Sci. Eng., 8, 4 (2006), 62-65.

[Najjar et al., 2003] Najjar, J., Duval, E., Ternier, S., Neven, F.: "Towards interoperable learning object Repositories: the Ariadne experience", Proc. IADIS Int'l Conf. on WWW/Internet 2003, I, (2003), 219-226.

[Namuth et al., 2005] Namuth, D., Fritz, S., King, J., Boren, A.: "Principles of sustainable learning object libraries", Interdisciplinary Journal of Knowledge and Learning Objects, 1, (2005), 181-196.

[Radenkovic and Markovic, 1999] Radenkovic, B., Markovic, A.: "An approach for specification of knowledge-based simulation models in the Web environment", Proc. Of TELSIKS 1999, Nis, Serbia (1999) 1, 313-317.

[Rossetti et al., 2009] Rossetti, M.D., Hill, R.R., Johansson, B., Dunkin, A., Ingalls, R.G. (eds.): "Motivaton of students for further education in simulation by an applied example in a related other course in engineering education – a case study", Proc. of the 2009 Winter Simulation Conference, Austin, USA (2009).

[Sampson and Zervas, 2008] Sampson, D. G., Zervas, P.: "Enhancing Educational Metadata with Science Education Information", Proc. of the 8th IEEE International Conference on Advanced Learning Technologies (ICALT 2008), Santander, Spain, IEEE Computer Society (2008).

[Stahl, 2001] Stahl, I.: "GPSS: 40 years of development", Proc. of the 2001 Winter Simulation Conference, Arlington, USA (2001) 577-585.

[Tao et al., 2006] Tao, Y.-H. Guo, S.M., Lu, Y.H.:"The design and the formative evaluation of a web-based course for simulation analysis experiences", Comput. Educ., 47 (2006) 414-432.

[Zhu, 2007] Zhu, X. H.:"Extending the SCORM Specification for references to the Open Content Object", Educ. Techn. Soc., 10, 1 (2007) 248-264.