

Dynamic Composition of Curriculum for Computer Science Courses

Zona Kostić
zkostic@singidunum.ac.rs

Aleksandar Jevremović
ajevremovic@singidunum.ac.rs

Irina Branović
ibranovic@singidunum.ac.rs

Dragan Marković
dsamarkovic@singidunum.ac.rs

Ranko Popović
rpopovic@singidunum.ac.rs

Faculty of Informatics and Computing,
Singidunum University
Danijelova 32, Belgrade, Serbia

Abstract— In this paper, a novel platform for curriculum development and design used for creating a Virtual University System is described. The platform has been developed based upon experience of using Web-based Computer Science virtual environment. The system consists of Interface, Automatic assessment and Tutoring modules, and is used for dynamic creation of new courses, syllabi, and curricula. The Web-based technology we applied enables the use of virtual environment for computer science courses in distance learning.

Keywords - *Dynamic curriculum; Virtual learning environments; Web-based education*

I. INTRODUCTION

This paper describes a novel approach to curriculum development and design in creating Virtual University System. Most existing educational environments use Virtual Reality (VR) techniques, which concern the creation and use of the Virtual Learning Environment (VLE) [1]. Use of the VLE at universities requires collaboration and interaction between the onsite and distant students and instructors.

Distance learning is an alternative and a supplement to traditional classroom instruction. The interactive nature of virtual classroom addresses the main challenges found in distance education, namely student involvement and participation. Virtual classrooms often rely on software simulators of pedagogical computer systems. A number of software tools targeting teaching and learning in introductory courses in computer science have been proposed and developed [2]. Also, dynamic curriculum development has been an active area of research [3] [4].

Our research goal has been the development of Virtual University System based on the experience of applying online teaching and learning for the last decade. Our virtual learning environment is designed to address the limitations of existing distance education systems. The result of our work is practically implemented system for distance learning and dynamic creation of curricula, whose novelty with respect to all current educational systems is threefold:

- Enables integration of virtual and real classrooms;
- Applies a specific pedagogical approach based on students' feedback;
- Allows for dynamic creation of new syllabi and curricula.

A curriculum defines learning content of a course or program of study in terms of knowledge and skills, i.e., specifies main teaching, learning, and assessment methods. Curriculum also indicates learning resources required to support the effective delivery of the course. A syllabus describes the content of a specific program of study and can be thought of as a part of curriculum [3].

The elements of a curriculum are: learning outcomes, content, teaching and learning methods, assessment, and virtual learning resources. There are a number of papers dedicated to curriculum development and design, [5][6][7][8]. None of them offers specific pedagogical approach based on student profiles and educational materials retrieved by using specific module, taking into account previous teaching experience. The curricula are developed and modified dynamically, through monitoring and evaluating at the end of each semester, when instructors compare them with corresponding curricula of the most renowned universities. Starting point for development is an IEEE curriculum standard in which the existing programs are dynamically modified or simple changes made to individual lessons [9]. The instructor chooses course materials based on his preferences and student feedback.

Creating new curricula is modular, aiming to provide instructors with some of the background theory related to curriculum design and course development, while integrating best-practice approaches and recent trends in computer science education. Curriculum development is an iterative process in three phases: evaluation, content modification, adding new methods and corrections.

Our Virtual University System applies the Problem Based Learning (PBL) methodology by implementing it as project-based learning [10]. This methodology engages students to integrate theory and practice, and see the big picture, rather than only pieces of the problem.

The analysis of system use, together with the observed fact that students tend to spend much time in virtual reality, justify the choices taken in development of a virtual system and orientation towards integrating virtual environment with reality.

The rest of the work is organized as follows: Section 2 contains background and motivation, concerns with technologies, environments, platforms, and related work; Section 3 describes phases and modules in dynamic creation of curriculum; Section 4 gives a case study example of

Computer Graphics (CG); Section 5 describes the results of the application and evaluation; and finally, Section 6 concludes and presents future work.

II. BACKGROUND AND MOTIVATION

There are two essential parameters relevant for developing virtual learning environments: the first are technologies used for implementation, and the second are previously implemented virtual environments upon which new ones can be based. This section gives a short overview of 3D technologies and environments related to education.

A. 3D Technologies

There are many technologies which enable the presentation of 3D data on the Internet; an excellent survey is given in [11]. The most commonly used are X3D (eXtensible 3D) and WebGL (Web-based Graphics Library), both designed for the creation of interactive Web-based and broadcast-based 3D content, and suitable to integrate with multimedia. WebGL works without installing additional software, but only within a compatible Web browser. Regardless of the fact that X3D works at much lower level and needs installation of an appropriate plug-in, it works within any Web browser, and as scene-graph system and with XML encoding, it is much better choice for beginning students.

B. Environments

Learning Management Systems (LMS) dominate in e-learning; the most prominent examples are Blackboard [12], Moodle [13], ATutor [14], and dotLRN [15]. These are integrated systems which support a wide area of distance learning activities. Platforms are often used in education and they are commonly divided into commercial and non-commercial products. For example, Sloodle [16] integrates Moodle, the open source tool for learning with Second Life, the most used commercial platform. Both commercial and non-commercial platforms lack important functionalities, such as cooperativity, real-life experience, and desktop sharing, while offering only average graphics quality. Most Virtual Universities are based on commercial platforms.

C. Related Work

There are many systems conceptually similar to ours, but none of them offers full range of functionality. In particular, these systems are not extendible. For example, the virtual classroom with smart tutor described in [17] is a good solution, but only for single courses. There are also solutions using X3D for creating virtual learning classrooms and labs. EVE [18] is the closest solution to ours, especially when it comes to CG course, but without real-time streaming. Paper [19] describes an excellent course for engineering students, but without collaboration, text or video chat. The solution [20] combines many features of X3D and ActiveX in creating virtual lab, but lacks the possibility of dynamic curriculum creation. Finally, [21] offers an excellent interface with much functionality, but lacks groupwork support. With respect to all described educational systems, our approach is superior because of the integration of virtual

and real classrooms, application of specific pedagogical approach based on students' feedback, and support for dynamic creation of new syllabi and curricula.

III. DYNAMIC CREATION OF CURRICULUM

Our Virtual University System consists of three software modules (Figure 1):

- Interface module
- Content retrieval module (MSearch)
- Assessment module (MTutor).

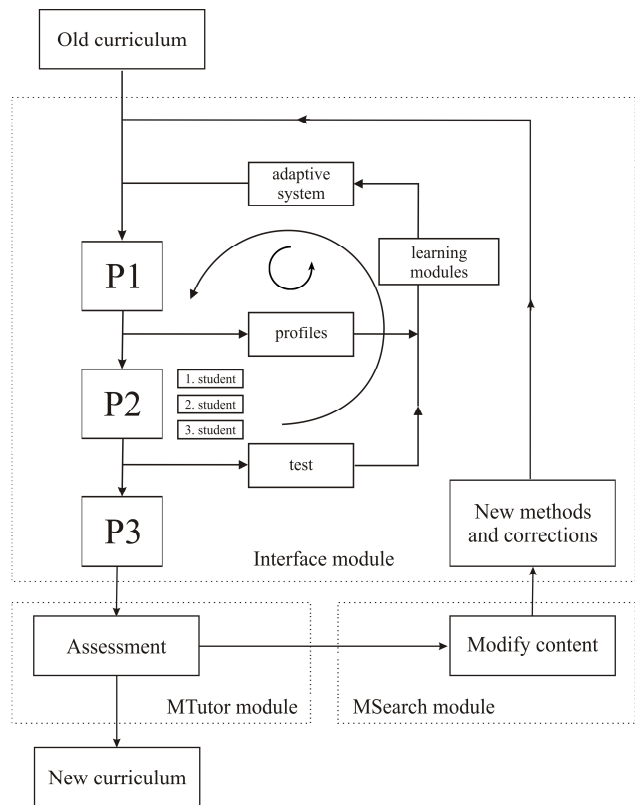


Figure 1. Three components of dynamic curriculum creation cycle.

Our platform allows for creating of new syllabi and curricula based on the previous ones. This process is dynamic; it iteratively uses student profiles and educational materials retrieved by using specific module.

We will begin by describing separate functionalities of three different modules, and proceed with an overview of dynamic curriculum creation process.

A. Interface module

This module defines the interaction between participants in the process, and serves as the starting point for assessment. Interface module is based on the following components:

- Component which defines student’s profile by using learning and adaptive modules;
- Component which defines the connection between a student and other group members working on the same project, teaching assistant and instructor;
- Component which defines the connection between instructors and teaching assistants.

Interface is not an independent component, but instead incorporates feedback from other modules. Specifically, the system is able to dynamically change the course contents based on student profile; instructor tracks the changes and selectively incorporates them into the syllabus based on his own judgment.

B. Modeling of virtual environment

Since most of computer science courses are taught in computer labs, the necessary component of our virtual environment is a virtual model of a real laboratory. A virtual lab is a component of interface module.

The basic components of any laboratory are virtual client computers. The physical interface takes the form of a classroom equipped with thin clients. By virtualizing client computers, important benefits such as flexibility and availability are achieved. Virtual machines are much cheaper and easier to install/clone than physical ones. Furthermore, they consume less electrical power and space. Also, because different labs are executed on core servers at different times, virtualization allows hardware reuse by switching between different virtual machines. The default protocol for accessing client virtual machines is VNC, making them platform independent. Software configuration of each virtual machine in a lab is different, customized for course requirements.

Other components of the lab include electronic educational materials (documents, simulators, and evaluation systems) and real-time streams. The laboratory core is built using cluster servers which provide execution of virtual machines, configured for educational processes.

C. Content module

MSearch retrieves educational material (in the form of pdf, ppt, and html files) from different universities’ Web sites based on user query for a text phrase and a query for an image name. MSearch allows personalizing the learning process, as well as reusing previous research efforts, results, and experience. At the end of each semester, teaching and testing materials are compared with other universities. This approach enables instructors to assess the difficulty level of their lessons and practical work with respect to the same courses at other universities, and to include other instructors’ experience in new syllabi and curricula.

MSearch searches Google Web and Google Images, with a query filtering set for academic domains. Dodget Get Links displays top 10 links for a text query and top 10 links for an image query (Figure 2). The top three ranked links (or a link of the user's choice) are passed to the crawler. Afterwards, complete indexing, retrieving images, as well as doc, rtf, and pdf documents from selected Web pages is performed.

Indexing of documents is multimodal, i.e., both text and image metadata are indexed.

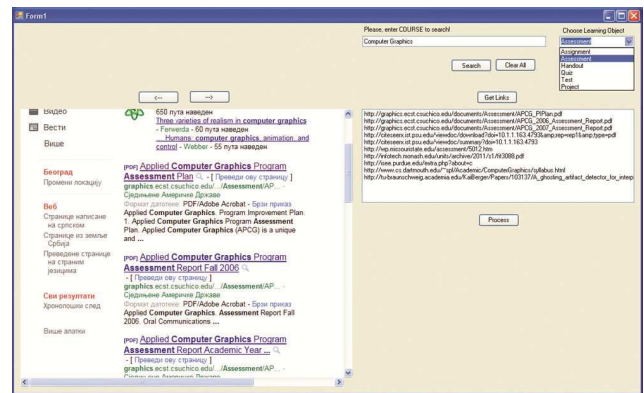


Figure 2. Text query and choosing learning object.

Based on the results that MSearch presents, the instructor is able to decide which link to use to compare retrieved exam questions with his own (Figure 3). The instructor can also define new questions or decide to import the questions retrieved from the selected link into the assessment module. This approach enables instructor to define the difficulty level of questions, and by taking into account students’ feedback he can balance students’ load accordingly. At the same time, the instructor incorporates changes of the curricula at universities that he considers relevant. For example, Figures 2 and 3 illustrate the use of materials from Berkeley, which is among top computer science universities.

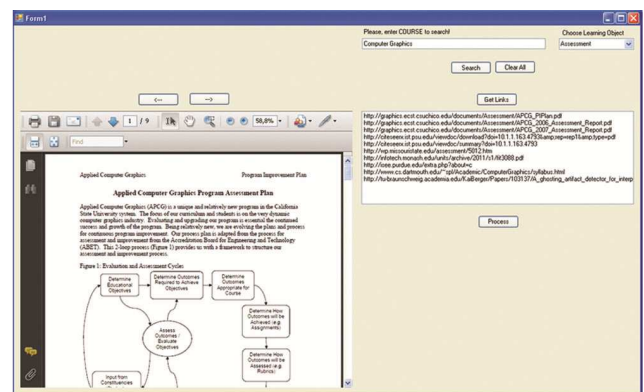


Figure 3. Text query results.

D. Assessment module

MTutor, is subsequently used for computer testing and defining appropriate learning steps by applying student-centered rules during learning and testing. MTutor system allows using textual and multimedia querying with true/false and multiple choice questions. Statistical processing of students' results is enabled on class, single student, question

and answer base levels. Besides grading students, these results are important for other purposes, such as recognizing high-quality questions (that are correlated with overall course goals) and eliminating low-quality questions. In this way, course questions pool is iteratively improved after every examination, and as a result overall validity and reliability of grading method used is improved. Another purpose is comparing student groups and class results by using integrated T-test component. This feature enables instructors to measure the impact of curriculum changes that were applied. Finally, statistical processing of students' results enables instructors to quantitatively measure performance of a single student during the course.

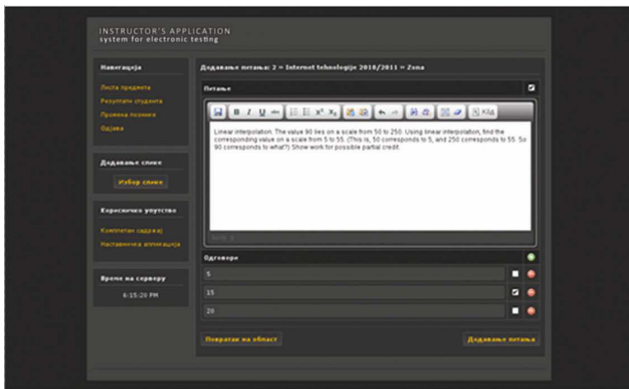


Figure 4. An example question in MTutor module.

The process starts and finishes with searching for various test types and comparing them with the existing ones using the automatic search engine and finishes by gathering the selected questions into the system for the final exam (MTutor module).

IV. PROCESS OF CREATING DYNAMIC CURRICULUM

Heterogeneous groups of students are formed based on students' learning styles. In general, adaptive systems for individual learning include modeled entities on which the decisions on adaptation are based (i.e., user preferences) described in [22].

Dynamic curriculum is developed in the following three phases (Figure 1):

- Phase 1: Pretesting, using the adaptive system and self-assessment to obtain student profiles and change the learning modules. After creating learning modules tailored to a specific student, heterogeneous groups of students are formed. A project is assigned to each group.
- Phase 2: The three subprojects are merged into the "big picture", followed by interactive assessment (supervised by an instructor) and exams, a final step in which everyone participates. The instructor makes final suggestions based on students' opinions about the work of others. The results of the second phase are documentation of the final project, with an option

to use different types of assessments, and adaptive testing.

- Phase 3: Objective assessment, using students' educational materials, is done as a final exam, under the responsibility of a supervisor. Student evaluation is based on this assessment. Upon conclusion of the exam, the supervisor corrects submitted materials and stores them. The supervisor and/or groups of students, based on the previous experience, proceed in making changes to learning modules of the old syllabus and iteratively change it.

Described phases are performed by using resources of a virtual lab, while the instructor is allowed to control and eventually alter their execution.

Course design and curriculum development tools are saved to be reused for subsequent iterations of the same process. The process of creating a dynamic syllabus is repeated for many courses and is eventually used for constructing a new curriculum. We use a pedagogical approach based on student-virtual system-instructor feedback. Each module contains an introduction to the topic along with explanations of concepts, illustrations, and, where appropriate, interactive components to visualize algorithms related to that topic.

Example: COMPUTER GRAPHICS LAB

We will explain the previously described ideas on an example in which creating a computer graphics lab is assigned as a group project to students.

IEEE curriculum [7] defines modeling, visualization, and virtual reality as principal interrelated fields of interest for a computer graphics course. These three main components were used for defining a computer graphics group project. The project is implemented through three subprojects assigned to each group; the subprojects are:

- Modeling 3D virtual laboratory environment (adding objects such as tables, PCs, routers etc.)
- Enabling communication by using predefined chat and real-time streaming components, and also connecting components created in the first subproject
- Integrating the two previously created components into a Web-based application.

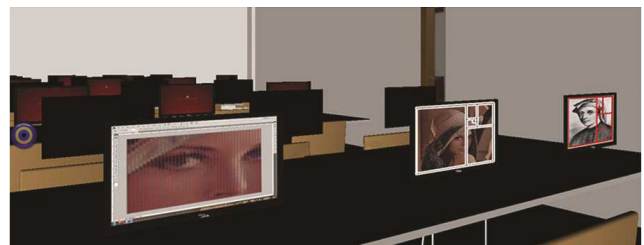


Figure 5. Implementation of the 3D lab for computer graphics course.

The first subproject consists of creating a 3D lab model in X3D with objects for computer graphics. Many tools for

3D modeling and animation enable exporting in X3D format; students used 3ds Max software for creating a final model of the lab (Figure 5).

The second subproject involves integrating virtual client computers into the graphics laboratory. The physical interface consists of thin clients. Web interface to client virtual machines is implemented as an X3D lab, with integrated Virtual Network Connection (VNC) client software. The idea is that 3D environment should replicate real laboratory.

Further development of the virtual lab is done in X3D-Edit by adding functionalities, such as interaction, animation, and navigation. Animation involves adding timers and interpolators to drive continuous events, interaction concerns mouse-based picking and dragging, while navigation defines user movements, collision, and visibility detection. It is also necessary to enable collaboration (synchronous communication) within X3D lab. A lab must contain a link to dynamic libraries, described in an XML file. A student must use Flash to connect to Virtual Machines.

V. THE RESULTS OF THE APPLICATION AND EVALUATION

In this section, we present a general evaluation of our virtual educational model and its implementation during two semesters. Qualitative and quantitative evaluation of the proposed educational approach was conducted. The qualitative evaluation included a number of student surveys and discussions with instructors. Surveys attempted to determine what students perceived as good educational tools and how they assessed the overall effectiveness of this approach. Students mainly complained about the steep learning curve of the system, and their suggestions have been implemented in subsequent iterations of our system. Few lessons at the beginning of each course were dedicated to training students on how to use the system. Also, manuals and tutorials explaining the toughest steps of using VUS were developed and integrated into the system. Manuals also describe how to use the existing tools in all different phases of learning, self testing, group work, and exam taking.

Students also complained that during preparation for self testing cannot decide which tools to use. This was also perceived as a difficulty in learning process; as a remedy, the option to choose the right tool based on experience of previous users (i.e., statistical data on most used tools and materials) was added. The statistical data are regularly updated upon completion of the course and used in a new syllabus.

Yet another students' complaint was the low speed of access to the environment. It had been noticed that in days before the exam the number of VUS site visitors had rose up to 80% of students. The problem of the access speed to the environment was solved by using the appropriate software for load balancing.

We also measured system efficiency by comparing final results of students who used the laboratories through the virtual and physical interface.

The exam was conducted in a controlled environment (Computer Graphics exam, Figure 5.) with 40 students, of which 20 used the physical environment, and the other 20 used the virtual environment. The testing process was supervised by two instructors and two teaching assistants.

The results of the statistical analysis (T-test) of final exam test scores are presented in Table 1.

TABLE I. RESULTS OF THE STATISTICAL ANALYSIS OF FINAL EXAM TEST SCORES ON A SAMPLE OF 40 STUDENTS DIVIDED INTO TWO EQUALLY SIZED GROUPS

	Control group (physical environment)	Experimental group (virtual environment)
No of students	20	20
Mean result (points)	71.55	79.35
Standard deviation	17.73	15.20
Variance	314.35	232.04

Comparing results with the corresponding values in the T-table (for a statistically acceptable p - value of 0.05, calculated on the basis of degrees of freedom for both groups) showed no statistically significant differences in results achieved by control and experimental groups.

The results of this statistical analysis, combined with the observed fact that students tend to spend much more time engaged in virtual reality than in using multimedia instructional materials, further encourage the orientation towards a completely virtual environment in the future.

The results of our analyses prove the usability of our approach and justify the attempts to dynamically adapt new curricula by comparing the results achieved by different generations of students.

VI. CONCLUSION AND FUTURE WORK

Our Virtual University System is developed as a distance education system to enable fully integrated real and virtual labs. The labs possess unique characteristics that address the limitations of existing distance education systems. New courses and their syllabi are defined using three software modules. The new Web-based curricula are created iteratively, taking into account previous teaching experience. The results of our experiments are promising enough to encourage further integration of Virtual University System with sensors, semantic 3D model retrieval, and 3D searching. In the next development phase, a significant research effort will be put into automating all phases in creating a Virtual University System by introducing artificial intelligence, through curriculum sequencing as a way of helping the student to find the best route through the educational material. The plan for future research is to integrate students' profiles as input parameters for curriculum sequencing to provide personalized learning paths through the course content (texts, exercises, examples, and questions).

ACKNOWLEDGMENTS

The work presented here has been supported by the Serbian Ministry of Education and Science (project grant no. III44006)

REFERENCES

- [1] C. G. Burdea and P. Coiffet, "Virtual reality technology", 2nd Edition, John Wiley & Sons, pp. 464, 2003.
- [2] K. E. Sanders and R. McCartney, "Program assessment tools in computer science: a report from the trenches", Proceedings of the 34th SIGCSE technical symposium on Computer science education (SIGCSE '03). ACM, pp. 31-35, 2003.
- [3] J. McKimm, "Curriculum design and development", School of Medicine, Imperial College Centre for Educational Development, pp. 32, 2007.
- [4] J. Brewer, A. Harriger, and J. Mendonca, "Beyond the Model: Building an Effective and Dynamic IT Curriculum", Journal of Information Technology Education, pp. 441-458, 2006.
- [5] T. Groover and J. Kabara, "The design and implementation of a pre-college computer science curriculum for underrepresented high school students", Frontiers In Education Conference: Knowledge Without Borders, Opportunities Without Passports, pp. T3A-22 - T3A-23, 2007.
- [6] J. Chookittikul and W. Chookittikul, "Six sigma quality improvement methods for creating and revising computer science degree programs and curricula," 38th Annual Frontiers in Education Conference, pp. F2E-15-F2E-20, 2008.
- [7] The Development of the IEEE/ACM Software Engineering Curricula
http://www.enel.ualgary.ca/People/yingxu/Publications/Papers/IEEE%20CR50_Wang.pdf (accessed April 6, 2012).
- [8] K. Georgouli, "Virtual Learning Environments-An Overview," 15th Panhellenic Conference on Informatics, pp. 63 – 67, 2011.
- [9] Computer Science Curriculum 2008
<http://www.acm.org/education/curricula/ComputerScience2008.pdf> (accessed April 6, 2012).
- [10] G. Simic and A. Jevremovic, "Problem-based learning in formal and informal learning environments", Interactive Learning Environments, DOI: 10.1080/10494820.2010.486685, 2010.
- [11] B. Turonova, "3D Web Technologies and Their Usability for The Project 3D Mobile Internet", Technical Report, Research and Development Center for Mobile Applications, Faculty of Electrical Engineering, Czech Technical University in Prague, pp. 18, 2009.
- [12] L. Ling and H. Lie, "Construction of Web-delivery Elaborate Courses Based on Blackboard: As an Example to the Course for Basic Circuit Analysis," International Forum on Information Technology and Applications, pp.287-289, 2010.
- [13] M. Amelung, K. Krieger, and D. Rösner, "E-Assessment as a Service," IEEE Transactions on Learning Technologies, pp. 162-174, 2011.
- [14] V. Gonzalez-Barbone, M. Llamas-Nistal, "eAssessment: Trends in content reuse and standardization," 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, pp.T1G-11-T1G-16, 2007.
- [15] D. Huerva, J. Velez, and S. Baldiris, "Adaption of Courses and Learning Environment to the User Context in dotLRN", Proceedings of the 2008 International Conference on Computational Intelligence for Modelling Control & Automation (CIMCA '08), pp. 1264-1267, 2008.
- [16] Z. Guomin and Z. Jianxin, "An Educational Value Analysis of SLOODLE-Based Distributed Virtual Learning System," Second International Workshop on Education Technology and Computer Science, pp. 402-405, 2010.
- [17] Y. Hu and G. Zhao, "Virtual Classroom with Intelligent Virtual Tutor", Proceedings of the 2010 International Conference on e-Education, e-Business, e-Management and e-Learning, pp.34-38, 2010.
- [18] C. Bouras, C. Tegos, V. Triglianios, and T. Tsiatsos, "X3D Multi-user Virtual Environment Platform for Collaborative Spatial Design", Proceedings of the 27th International Conference on Distributed Computing Systems Workshops, pp. 40 – 40, 2007.
- [19] P. Goeser, W. Johnson, F. Hamza-Lup, and D. Schaefer, "VIEW: A Virtual Interactive Web-based Learning Environment for Engineering," IEEE Advances in Engineering Education Journal, Special Issue on Research on e-Learning in Engineering Education, pp. 24, 2011.
- [20] S. Liang and P. Rong-jiang, "A 3D virtual experiment framework based on X3D and ActiveXL", 2010 International Conference on Audio Language and Image Processing, pp. 1035–1039, 2010.
- [21] Haiqing, Y. Guofu, and F. Jie., "Research on the Collaborative Virtual Products Development Based on Web and X3D", Proceedings of the 16th International Conference on Artificial Reality and Telexistence, pp.141-144, 2006.
- [22] S. Marković, N. Jovanović, R. Popović, and A. Jevremović, "Adaptive Distance Learning and Testing System", Computer Applications in Engineering Education, DOI: 10.1002/cae.20510, 2010.