2010 14th Panhellenic Conference on Informatics

Web-Based Simulation Systems in Technological Education Implemented as Micro-Worlds

Anastasios Oikonomidis
School of Technological Applications, Department of Automation
Technological Education Institute of Piraeus
Greece

Constantinos Iliopoulos
School of Computing
University of the West of Scotland
United Kingdom

Abstract: The use of simulation technologies has become more frequent in recent years for many engineering instructional laboratories. The proliferation of high Internet connection speeds and the emergence of promising Web technologies such as AJAX have created opportunities to build simulation applications that are fully accessible via browser applications. The combination of such technologies with Web graphics languages such as SVG, could lead to immersive and graphics rich Web environments which could be used for the creation of educational micro-worlds and virtual laboratories. Such simulated educational environments would enable on and off-campus engineering students to work on simulated procedures, experiments and exercises similar to the ones normally run in face-to-face classes, and in this way accrue multiple educational benefits.

I. Introduction

Instructional laboratories have a significant role in the engineering education process. According to Lyle D. Feisel and Albert J. Rosa [1] the purpose of the engineering profession is to "manipulate materials, energy, and information, thereby creating benefit for humankind. To do this successfully, engineers must have knowledge of nature that goes beyond mere theory-knowledge that is traditionally gained in educational laboratories". In order to fulfil people's expectations, every engineer needs to be able to practice what he or she was taught in theory and for this reason laboratories operate in every technological institute providing a means for acquiring hands-on experience. This kind of experience is acquired through the execution of scientific experiments on small scale models, simulating the conditions of a real production environment. A key gain is a deeper understanding of a possible range of outcomes in real and bigger scale series of situations.

With the emergence of information technology, laboratories changed significantly. In some cases they were changed completely. A digital computer can facilitate - besides data storage - data analysis and presentation. The use of computers also helped researchers of other disciplines like mathematics, to provide better and more accurate models to engineers. As models were getting more and more accurate, the idea of creating software to simulate the real experiments attracted many enthusiasts. That trend lead to the creation of software which can simulate an experimental procedure in a way very close to the real thing. One of the most successful pieces of software for electronic circuit simulation was developed in the late 1970s at the University of Berkeley USA, by Ron Rohrer and his student Larry Nagel. That program was SPICE, which is an acronym for Simulation Program with Integrated Circuit Emphasis.

Nowadays it is well understood that simulation software can save money. Instead of purchasing or developing many different experiment kits for student needs, instructors can design the experimental process as a simulation in a relevant software application and pass it to their students along with the appropriate procedure/s. In a typical laboratory environment of the real world, there are working stations. Each one of them could consist of instruments like oscilloscopes, multi meters, function generators and other equipment necessary for the experiment procedure, like electronic circuits, motors, tubes, tanks and many others. All of them form a micro world which is built to facilitate a specific purpose: the education of students on one or more scientific procedures. So, is it possible to formulate a similar micro-world by using software tools so that students are able to perform the experiments on a personal computer? Of course at this point several additional questions arise:

Can we be sure that models used in simulations are accurate? Can simulations provide the same level of experience acquired in the real world especially for undergraduate students who are likely not to have much experience? If students having a computer at home can perform the experiments by themselves, what is the need of an instructor? Is simulation really cheap since many of the software vendors sell their products at high price, which can get even higher as many licenses per laboratory are needed?

Emerging technologies like Asynchronous JavaScript and XML (AJAX) and Scalable Vector Graphics (SVG) are likely to be used in order to create micro-worlds suitable for educational

978-0-7695-4172-3/10 $26.00 © 2010 IEEE
DOI 10.1109/PCI.2010.36

purposes.

In the following sections the role of simulations in engineering education, the features of SVG and its advantages over other competing technologies, and finally simulation application needs are discussed. Next, the presentation and brief discussion of an existing simulation application - micro-world follows.

II. Simulation In Engineering And Technological Education

Naps et al indicated in a recent study (2003) instructors believe that the use of visualization techniques help students learn. The major expectation is the enhancement of the learning process by improving student motivation; make both teaching and learning "more fun" after the integration of visualizations into the course materials [2]. In addition, especially when referring to engineering students, there is a great need for visual communication of their ideas both to "peers and the general public" according to the participants of the interdisciplinary workshop about "Visual Learning for Science and Engineering" [3]. One of the most common ways to implement visualization is by using simulation software. As personal computers become cheaper and stronger, the idea of using special software to simulate processes of the real world into classrooms and laboratories is supported in many institutes, like the U.S. National Science Foundation[1]. Of course, like every innovation, the use of simulation software takes serious criticism by sceptics like Professor Nick Bostrom of Oxford University[2] who point to problems such as students lacking of real hands-on experience. Arguments by the two opposite sides lead to the need of addressing the question whether simulation is really needed or not and if the answer is yes, to which extend.

According to Morgan and Jones [4] engineering students can benefit in a number of ways if the use of computer simulation software is included in the curriculum of the school they attend. One of the benefits is the motivation of students who actually use simulation software to enhance their learning. Science and engineering students usually need a strong theoretical base which involves extensive knowledge of mathematics. Engineers are expected to apply theoretical concepts to everyday problems and therefore act as problem solvers. In order to achieve that, they need first to acquire practical experience by experimenting on real world processes. The usual way, and for many decades the only way to get hands-on practice, is by working in laboratories equipped with the appropriate instruments and other apparatus in order to perform experimental procedures. Sometimes, unfortunately, these laboratories provide limited access to students either because of the small number of staff members or the large number of students, or both[3]. Attendees are forced to work hard and fast, collect the required data on time before the next group of students appears asking for their turn to work on the real world models that are usually found in engineering laboratories. Those leaving, frequently wish they had some more time at their disposal in the lab. This obstacle can be managed if there are one or more members of specialized staff present at the lab during days of the week when there are no scheduled classes, and therefore equipment is available for students to review previous experiments. Even if there is such a facility provided, it is possible that the number of students willing to use the same instruments at the same time may exceed the number of available instruments. This problem could be solved by providing more instruments to satisfy the needs of all users. Unfortunately, the cost of lab instruments is high, sometimes extremely so. Purchasing additional equipment to cover the needs of every student, while at the same time paying for extra working hours of staff, increases the operational costs significantly. One way to resolve this problem is by using computer based tools to simulate experiments. Kheir, N. A., et al [5] argue that "as computers and control design software become more readily available and more powerful, these tools will be ever more widely used to give a sense of more and more realistic experiments to each student. The result will be better prepared control engineers". They also suggest that in a laboratory environment "a personal computer can be used to implement the control algorithm, for graphics, man-machine interfacing and for computer aided instruction". Of course a personal computer can implement not only a control algorithm but any algorithm regardless of the discipline it comes from, since an algorithm is a mathematical expression and mathematics is what computers can do best. If there is simulation software available which is capable of reproducing experiments, procedures or exercises on screen then students can repeat them as many times as they need or like and even run them before putting their hands-on the real thing, and therefore can go to class well prepared. From the instructor's point of view, the use of simulation tools can help "speed up the teaching, analyze the problems more deeply, and make the arguments exhaustive" and all of these in combination with the Internet as "a channel to reach the learners at their homes" as Grimaldi and Rapuano [6] state.

The cost factor cannot be ignored in respect of simulation tools. Acquiring as many software licenses as the students attending a course may not be an option due to high prices. Fortunately, this is not always necessary since agreements between academia and software producers exist to make software tools available to those needing them at reasonable price rates. Besides that, companies often offer their products in student versions which are much more affordable than the professional ones. Additionally, open source tools may offer an absolutely free of charge solution to those interested in them. Wikipedia provides a list[4] of computer simulation software to its users where almost half of them are open source or at least give free licenses to students.

The above initial review indicates that using simulation software tools in engineering benefits both students and universities and therefore their use is for the best interest of both parties.

1 http://spectrum.ieee.org/energy/fossil-fuels/power-education-at-the-crossroads (1 Dec 2009)

2 http://www.simulation-argument.com/ (1 Dec 2009)

3 http://www.nap.edu/openbook.php?record_id=589&page=76 (1 Dec 2009)

4 http://en.wikipedia.org/wiki/List_of_computer_simulation_software (24 Feb 2009)

III. Micro-Worlds As Educational Platforms

According to Iliopoulos [7] micro-worlds when compared with conventional simulation environments exhibit "more comprehensive mimetic properties" while they "encourage participants to immerse themselves into complex and virtually realistic worlds". Micro-worlds can be founded as 3D virtual environments which "may afford pedagogical support for fostering constructivist learning environments, for geographically distant learners, because they provide educators with an accessible means of creating a rich and compelling 3D context for situating learning, communicative tools to support discourse and collaboration, and Web integration to provide just-in-time resources and information-seeking tools", according to Michele D. Dickey [8]. The term constructivist comes from the belief that "knowledge is constructed, not transmitted, and that learners play an active role in the learning process" [8]. In a 3D educational environment or in general in a graphics-oriented online environment multiple users can interact and participate by exploring it. They can also communicate with each other and the instructor, and use its provided functions; this kind of environment is likely to "foster the development of problem-solving skills" [8]. As was mentioned earlier, engineers are supposed to be problem solvers so the use of micro-worlds as tools to develop this ability can prove to be really helpful.

IV. Svg As Platform For Creating Micro-Worlds

Sir Timothy John Berners-Lee inventor of the World Wide Web and director of the World Wide Web Consortium (W3C) which "oversees the Web's continued development"[5] expressed in the late 90's[6] his vision about the evolution of the Web which is known as the concept of the Semantic or Second-generation Web. Semantic Web is also defined as "the extension of the World Wide Web that enables people to share content beyond the boundaries of applications and web sites"[7]. Content sharing reveals the need for specific description of data so that it can be easily handled both by humans and computers in a platform independent fashion. Traditional hypertext mark-up language (HTML) is not adequate on its own for this evolution since it was designed to present data, not define it. The introduction of the extensible mark-up language (XML) in 1998 has offered to the developers' community a language to define data through self-describing tags that allow easy interpretation. With the use of XML, content can be separated from its presentation rules while an XML document is purely text and therefore can be used in a large variety of presentation devices including mobile phones. Virtual learning environments for engineers demand the use of a variety of different kinds of data like text, mathematical formulas, images, charts and in many cases audio and video files for supporting scientific concepts and to provide the sense of reality. The Semantic Web offers the opportunity for all these kinds of data to be handled by the use of XML based languages and delivered to end users regardless of the platform or the browser they use. A web site can therefore be considered more as a web application serving the needs of each user independently rather than serving just the same static content to everyone. As it was mentioned earlier, information visualization is very important in a virtual learning environment. Especially, scientific concepts can be much more easily understood by the use of graphics related to them [3]. Plain figures are likely to have little meaning to a big number of students. For example, the change of liquid level contained in a tank can be either represented by numbers or by a combination of interactive graphics depicting the tank, its container, the container's level changes, and finally numbers. In order to achieve such a graphic representation and deliver it over the Web a language different than HTML is required. Scalable Vector Graphics (SVG) is such a language. According to Cagle [9] HTML uses a "basic, open, declarative vocabulary, (by which) one can describe a comprehensive subset of all hyperlinked documents" while SVG "takes this same underlying concept and applies it to the realm of vector graphics". SVG is a language that describes two-dimensional and graphical applications in XML, while it is a W3C recommendation [10].

Graphical entities can be identified with different names related to entity use. HTML allows an identification parameter to be added to all tags and therefore makes referencing to them an easy task; the same approach is possible with SVG. Other graphical languages allow naming on groups or layers, while in SVG every graphic element can be named and controlled individually. Text can be presented in accordance with the design requirements. With SVG text is easily handled not only in single lines but also in any artistic way like across any shape, while Cascading Style Sheets are used the same way as in HTML. Through this approach text and graphics are not separated the way they are in photographic images like Joint Photographic Experts Group (JPEG) or Graphics Interchange format (GIF) where text makes no difference and is treated as any other kind of graphics.

There are many ways to interact with SVG. A complete and native Document Object Model (DOM) exist in SVG which is the same as in HTML. This allows graphic elements' manipulation to be undertaken by many high level programming languages. The most commonly used is JavaScript while there are SVG processors allowing C++, Java, Perl, PHP and others to handle the SVG DOM as well.

The creation of graphics libraries is easy. With SVG the linking between graphics is possible. This means that within an SVG file, content from other parts of the file or even other files can be linked and therefore the implementation of libraries containing graphics becomes possible.

Data is easily described. Metadata, which is also known as data about data, is essential for the evolution of the Semantic Web. Data that is properly described can be easily found and handled. Since SVG is XML based, metadata can be embedded within SVG files. Thus, entities of SVG are possible to be described inside the SVG file and then get bound with external metadata. This characteristic makes SVG the most "self-aware"

5 http://en.wikipedia.org/wiki/Tim_Berners-Lee (9 May 2009)
6 http://www.w3.org/DesignIssues/Semantic.html (9 May 2009)
7 http://semanticweb.org/wiki/Main_Page (9 May 2009)

language among other graphical languages like Postscript [9]. SVG is recommended by W3C up to its version 1.1 while efforts for a new version 2.0 are undertaken, in order for new specifications to be provided, and a new SVG framework to be created, which will work on a comprehensive range of devices and platforms [10]. SVG is also recommended by W3C for its version Tiny 1.2 targeting mobile devices. Today most Web browser vendors have included SVG support into their products.

V. Advantages Of Svg Over Other Competitive Means Of Visualization

SVG, as it is stated in its definition[8], describes two-dimensional vector graphics[9] and since it is XML based an SVG file can be created and edited even with the simplest text editor. While raster graphics[10] are formatted as a rectangular grid of pixels, in vector graphics mathematical formulas are used to convert data to image either on screen or on paper. Mathematical formulas are also used to scale or stretch graphics. Because raster graphics contain information for every single pixel like its color and consist of the exact number of pixels they occupy in a computer (or other device) screen they cannot be scaled without quality loss, while vector graphics are rendered on screen by software translating data to pixels, and therefore they don't lose any of their sharpness when they are enlarged. Raster graphic files are characterized by their resolution measured in dots per inch (DPI) or per centimeter (DPC). Usually, the higher their resolution is the better their quality but this requires bigger files, while vector graphics files are usually much smaller for the same size. Raster graphics are better for complex pictures while vector graphics are better for representing technical drawings like mechanical or electrical ones. Both vector and raster graphics file formats are supported by Web browsers. Commonly supported raster file formats are GIF, JPEG and Portable Network Graphics (PNG). While raster graphics cannot be scripted, this is much easier with vector graphics especially with SVG which is XML based. Although raster images can be embedded into vector graphic files it is better to use raster and vector files separately according to their intended use.

Besides SVG, other vector graphic file formats exist. Most popular of them and most widely used is Adobe® Flash™ (formerly Macromedia® Flash™). Many claim that Flash™ is a competitor of SVG. Sara Porter argues this is not a case [9]. When pure animation is required then Flash™ works much better than SVG where hard work on scripting is needed for the same result. Porter states that although SVG lacks in animation it is much more scriptable and flexible than Flash™ is. The XML characteristics of SVG allow it to be integrated with complex databases and convert it into a powerful tool. Flash™ is great for creating advertising clips while SVG can be used for more serious purposes [9]. SVG is a W3C standard while Flash™ is not and no browser supports it natively. Many tools support SVG while Flash™ is developed by only one vendor (Adobe®), and expensive software tools are required to create Flash™ files. SVG files are open and can be compressed occupying even less space while Flash™ files (using an swf extension for "Small Web Format") are usually larger[11]. The creation of SVG libraries is possible, consisting of many symbols which are easy to be used by other SVG files, something that is not possible with Flash™ files. Text search is easy with SVG and Web search engines can locate text within SVG files, which is very important with the semantic Web. On the other hand, Flash™ files leave no such potential to search engines. Scripting with SVG can be performed by JavaScript and other languages with the DOM allowing a full event driven client side scripting, while in Flash™ this can only happen with ActionScript which is an implementation of ECMAScript standard, but "in a different programming framework with a different associated set of class libraries"[12]. Other areas that SVG excels over Flash™ is styling through CSS, embedding XHTML, and compatibility with the XML three dimensions standard also known as X3D, which is used for 3D representations.

Another SVG competitor is Microsoft® Silverlight™ which is "a programmable web browser plug-in that enables features such as animation, vector graphics and audio-video playback that characterizes rich Internet applications"[13]. Although with Silverlight™ text is represented in a way that "is searchable and index able by search engines as it is not compiled"[14], it is not as open a standard as SVG is. With Silverlight™ "user interfaces are declared in XAML and programmed using a subset of the .NET Framework"[15]. XAML is the acronym for Extensible Application Mark-up Language, which is a declarative XML based language created by Microsoft®. Because it is XML based it allows developers to share content among them freely and compilation is required. Also, no complex tools are required to process XAML. Yet, "as it is strongly linked to the .NET Framework 3.0 technologies, the only fully compliant implementation as of today is Microsoft's"[16]. It is surprising that Microsoft® invented a new way to represent vector graphics while there was already one. Microsoft® seems that doesn't care at all about SVG[17] since there is no SVG support in all versions of Internet Explorer™. Technologies that are closed and bound to a single organization leave little space for others to make use of them. In the end,

8 http://en.wikipedia.org/wiki/Scalable_Vector_Graphics (17 May 2009)
9 http://en.wikipedia.org/wiki/Vector_graphics (17 May 2009)
10 http://en.wikipedia.org/wiki/Raster_graphics (17 May 2009)

11 http://www.moock.org/webdesign/svg/articles/svg-featureset.html (17 May 2009)
12 http://en.wikipedia.org/wiki/Adobe_Flash#Programming_language (17 May 2009)
13 http://en.wikipedia.org/wiki/SilverLight (17 May 2009)
14 http://en.wikipedia.org/wiki/SilverLight#Overview (17 May 2009)
15 http://en.wikipedia.org/wiki/SilverLight#Overview (17 May 2009)
16 http://en.wikipedia.org/wiki/Extensible_Application_Markup_Language#Overview (17 May 2009)
17 http://weblogs.asp.net/jgalloway/archive/2007/06/05/silverlight-and-xaml-have-you-guys-met-old-man-svg.aspx (17 May 2009)

many different technologies doing the same thing open the road for the Internet to become a "Babel's tower"; this approach goes against the efforts of W3C for "building consensus around Web technologies"[18]. This is not the case with SVG which is currently the only open and most widely supported vector graphics standard.

VI. Specification Of The Application Needs

A virtual micro-world simulating the working spaces of a real world engineering instructional laboratory should look like them as much as possible. SVG related technologies can be very helpful in achieving this objective, since it is possible with SVG to reproduce a variety of scientific instruments, in terms of appearance and functionality. Through this graphic representation distant users will be able to view the same instruments and laboratory apparatus as if they were in front of them physically. Furthermore, the display panels on each virtual instrument should present numbers or graphs the same way the real ones do.

Also, users should be able to interact with the micro-world and to handle moving parts like knobs, switches or rotating buttons in a way as close as possible to the real ones. Indications like lamps (in ON-OFF conditions) should change status in a clear and visible way, making obvious to distant users that something has changed.

The next figure (figure 1) gives an example of a digital oscilloscope as it is found in many laboratories:

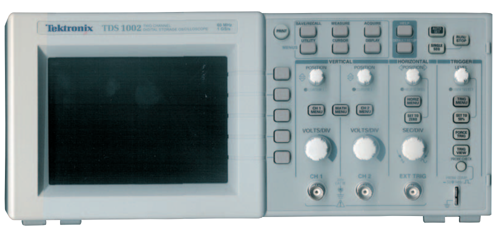

Fig. 1. Image of a real digital oscilloscope

In the next figure (figure 2), the previous oscilloscope's simulated counterpart is depicted for comparison:

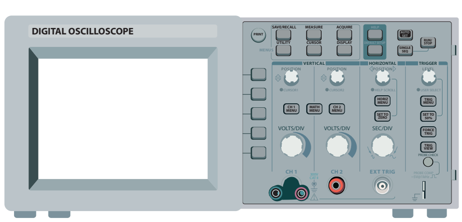

Fig. 2. Image of a simulated oscilloscope using SVG

By replicating carefully the real instrument image, the result is possible to look pretty much the same as the original one.

Since it is possible that not all engineering students are computer enthusiasts or experienced users, the application should run on a Web site which is accessible by students throughout the whole of the academic session. Including additional valuable information and services, not only regarding laboratory activities, but also about theoretical concepts being taught elsewhere (e.g. by the supervising professor) represent additional ways of enriching the micro-world. Typically, the lecturing part and the applied part are two different modules in engineering modules and students are examined and awarded grades and marks separately for each one of them. Therefore, the application could be part of a web site assisting students and providing information about different but yet relevant modules and module activities. Students will therefore find themselves using a familiar web site to conduct their online experiments when required. The whole Web site should run smoothly on different browsers and operating systems and provide its services in a clear, fast and user friendly way with plain and neat aesthetics, helping users to locate easily what they are looking for. Also, extra care should be taken for compliance with accessibility rules, and W3C recommendations.

Extra care should be taken in order to address student educational needs. For this reason a teaching model should be used as a guideline for the application design and implementation. One such model is that of Richard Felder and Linda Silverman [11] which is focused on teaching engineering students. This model suggests that the content to be taught should be related with other already known subjects and connected with students' personal experiences, so that students can have a holistic view of the course, which is likely to increase learning motivation. There must be a balance between the theoretical concepts that are taught through the application and the practical tasks in order to attract both kinds of students who prefer theory or practice. The simulation elements of the web site should perform as close to reality as possible so as to let students feel that they are not working on something different than what they have experienced in real life. Sufficient help and guidance should be provided, accompanied by charts and images where appropriate, in order to keep users away from dead ends. All steps should be stated in a clear manner, allowing short intervals for thinking about what has been completed so far and what is about to come next. The use of a user log-in mechanism can help assign students into working groups and thus promote cooperation between them. In addition, a range of encouragement and reward mechanisms, such as virtual applauding, could be used in the micro-world in recognition of a successfully completed task by a student.

VII. Application Presentation

A web site has been developed in an effort to implement the previously discussed guiding principles. It can be accessed at http://auto-hsae.teipir.gr. The web site currently serves the educational needs of the students of the Automation Department of the Technological Education Institute of Piraeus, Greece. It deals with both the theoretical and practical concepts of the Automatic Control Systems module. A key benefit for users is that they can perform simulated experiments in a graphics-rich environment, similar to that of the real laboratory, in terms of appearance and functionality. In order to access the micro-world, students (or anyone else wishing to) will need to open the "Laboratory" list and then click on the "Simulation" element. A short description

18 http://www.w3.org/Consortium (17 May 2009)

informs users about the application and advises on browsers to be used. Upon application start, users are in front of a micro-world consisting of vector images of a digital oscilloscope, a function generator, a digital multi-meter, a board known as "analog computer" including electronic circuits and a box holding cables.

In the real laboratory students have on each working station exactly the same pieces of equipment. They use cables in order to connect the different electronic circuits between them and with the instruments and they view the output either on the oscilloscope or the multi-meter screen. Students are asked during each semester to perform eight projects, each one of them consisting of various different experiments and by this procedure they apply theoretical concepts. There is at least one mathematical formula for each concept. During the design part of the micro-world a unique identification tag has been assigned to every different part of it. By scripting these parts with JavaScript, user events like mouse click are assigned to various parts and different features are implemented through functions. Such features are the possibility of rotation (simulating the rotation of a potentiometer), change of color (simulating status change) etc. Students are asked to click on a menu within the application, so as to initiate the procedures related to a specific simulated experiment. There is an array holding all the possible correct connections for each of the implemented experiments. After clicking on a menu selection, the theoretical diagram (also known as analog diagram) of it is presented on the oscilloscope screen. Students should click on the right connectors and "cables" move from the "box" in order to be attached on them. Actually the graphical representations of cables are SVG lines operating as entities where their attributes of XY coordinates change so as to give the impression of applying connections. When students decide that they have finished with all the necessary connections, then they run the simulation in order to view the result on the oscilloscope screen, as it is shown in the following figure (figure 3):

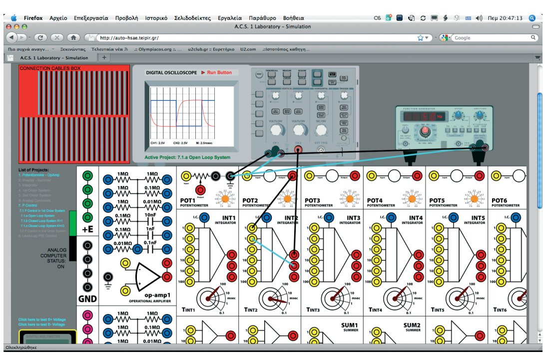

Fig. 3. Connection and result on oscilloscope

The application compares user moves with all possible correct connections stored in a related array and if there is a match it uses the mathematical formula as an algorithm to calculate results. Those results are converted to XY coordinates and passed as attributes to two poly-lines (one as system input and the other as output) which are rendered within the oscillator screen box.

VIII. Conclusions

Students who have used this micro-world have expressed their overall satisfaction with the application, mainly through anonymous questionnaires and direct, verbal feedback. The micro-world has been used since the Fall semester of 2008 to augment student learning. A considerable number of students have requested that the micro-world is expanded further to include simulations of all experiments they are asked to perform in the real-world laboratory. Students have also highlighted, either through questionnaires or in class conversations, the importance of the micro-world in helping them understand the concepts which are applied and tested through the various experiments and exercises, by enabling them to repeat them in their own time and for as many times as necessary. Students have also argued that the application is significantly user friendly, since it looks and operates pretty much the same as the real instruments that they already know how to use. They have also compared this micro-world with other simulation applications and have found that their complexity often impedes their learning. Conversely, students have indicated that they know how to navigate and what to do with this micro-world from the very first time of using it. Therefore, instructors need to introduce the micro-world very briefly.

Further work is required to expand and enrich this micro-world and evaluate its strengths and weaknesses more thoroughly. The development and deployment of the micro-world so far has demonstrated the technical challenges in developing micro-worlds but also their potential to meet the needs of learners and educators.

References

[1] Feisel, Lyle D. and Rosa, Albert J. The Role of the Laboratory in Undergraduate Engineering Education. Journal of Engineering Education. January 2005, Vol. 94(1), pp. 121-130.

[2] Evaluating the Educational Impact of Visualization. Naps, Thomas, et al. Thessaloniki : Annual Joint Conference Integrating Technology into Computer Science Education, 2003. pp. 124-136.

[3] Barry, Ann Marie, et al. Visual Learning for Science and Engineering. *ACM SIGGRAPH Education Committee.* [Online] ACM SIGGRAPH Education Committee, June 2002. [Cited: March 7, 2008.]

[4] Morgan, Roger and Jones, Karl O. The use of simulation software to enhance student understanding. *Engineering Education: Innovations in Teaching, Learning and Assessment (Ref. No. 2001/046), IEE International Symposium on.* Liverpool : s.n., 01 05, 2001. Vol. 2, pp. 33/1-33/6.

[5] Kheir, N. A., et al. Control Systems Engineering Education. *Automatica.* 1996, Vol. 32, 2, pp. 147-166.

[6] Grimaldi, D. and Rapuano, S. Hardware and software to design virtual laboratory for education. *Measurement.* 09 03, 2008, Vol. 10, 1016.

[7] Iliopoulos, Constantinos. Enabling Knowledge Activities, Learning And Systemic Thinking In Organisations: On The Relevance Of Simulations And Microworlds. *Systems and Knowledge Management - Special Edition of Systemist.* 2003, pp. 1-11.

[8] Dickey, Michele D. Teaching in 3D: Pedagogical Affordances and Constraints of 3D Virtual Worlds for Synchronous Distance Learning. *Distance Education.* 2003, Vol. 24, 1, pp. 105 - 121.

[9] Geroimenko, Vladimir and Chen, Chaomei, [ed.]. *Visualizing information using SVG and X3d:XML-based technologies for the XML-based web.* London : Springer-Verlag, 2005.

[10] W3C SVG, Working Group. World Wide Web Consortium - Web Standards. *Scalable Vector Graphics (SVG).* [Online] World Wide Web Consortium, April 15, 2009. [Cited: May 12, 2009.] http://www.w3.org/Graphics/SVG/.

[11] Felder, Richard M. and Silverman, Linda K. Learning And Teaching Styles In Engineering Education. *Engr. Education.* 1988, Vol. 7, 78, pp. 674–681.