AUTHORING-TOOLS FOR WEB-BASED SIMULATIONS

C. Brigas, M. J. Marcelino

ABSTRACT

With the appearance of Internet and its applications to education, great interest was developed in supporting learning by this new medium as it offers cheap delivery of materials, easy and comfortable access, etc. With all the possibilities that Internet gives us the idea of integrating simulations into Web pages and materials soon came up. However, simulations are time-consuming, difficult and expensive to produce, especially for the common teacher and without using the suitable tools, as most products for Web page development do no possess the primitives to build simulation items easily. In recent years, several tools have been developed to support the construction of simulations to integrate into Web-based materials. In this paper we review, analyse and contrast some of these tools. As a result of this study, and because we think that this is an area where there is still a lot to be done, we present a set of criteria that we believe Web-based simulation tools should conform to in order to support the work of the author of Web-based educational simulations.

KEYWORDS

Authoring-tools, Web-based Simulation, e-Learning

INTRODUCTION

Computer based-simulations have been used in many ways in science education for already some decades and its value and merits are well known and documented (Alessi and Trollip, 1985). By simulation we mean a software program that represents a situation in a dynamical way, usually described by a model, with which the learner can interact (Marcelino, 1998).

In spite of the benefits a simulation can bring us, they are time-consuming, difficult and expensive to produce, especially for the non-technical teacher and if the proper tools are not used. As a result, various products have been developed over the years to support the author in this task and automate totally, or at least partially, the development process (Marcelino, 1998, Jong and Sarti, 1994).

The pressure imposed by users of both simulation and the Web, along with the usual difficulties associated with simulation building, has lead to the necessity of implementing simulation tools for the Web, as most products for page development do no possess the primitives to build simulation items easily and the old tools (for non-Web environments) are not appropriate anymore. The union of these two services in the same tool, simulation and the Web, allows a very powerful product, with enormous advantages, among which we can mention: easiness of access, sharing of information, standardization, independency, etc. (Kuljis and Paul, 2000). In recent years some tools have appeared that allow integrating, in a more or less automatic way, simulations into Web pages.

The objective of this article is to analyse some of the current available tools in the market, identifying common characteristics, advantages and disadvantages, and to propose a set of criteria for Web-based simulation tools.

TOOLS FOR WEB-BASED SIMULATION

The application of Web-based educational simulations has already been done successfully during the last years (Atolagbe and Hlupic and Taylor, 2000, Bravo and Ortega and Verdejo, 1999). The number of available applications to build Web-based simulations is still small, however. Among them we find JavaTHESIS[1], Multiverse eSim Builder[2], Easy Java Simulations[3] and AgentSheets[4] that will be analysed herein. We will start by describing each of them in some detail. After, a comparative table with several evaluation criteria will be presented along with some comments.

JavaTHESIS

JavaTHESIS is a tool that allows the development of simulations of dynamic models. Dynamic models are models described by differential equations with the time as the independent variable. JavaTHESIS builds on the concepts of an earlier tool, the MacTHESIS[5] system. JavaTHESIS is based on Java-applets and the SimLib library.

The construction of a simulation is accomplished by altering a typical simulation program (Cascade). The author of the tool divides the alterations to be effected in two groups, HTML code and Java code.

In the first group it is necessary to configure the parameters of the applet and of the page where the simulation will be integrated and visualized. In the second group, it is more complicated as it is necessary to introduce the variables and the equations that represent the model to be simulated and that the author's tool identifies as the tool degrees of freedom.

In a JavaTHESIS simulation there can be two types of parameters: input parameters and output parameters. These parameters can be represented through scrollbars, graphs or counters.

The JavaTHESIS tool was developed for users with varied levels of knowledge of the Java programming language. However, whenever it is necessary to modify a model parameter it is necessary to have knowledge of both Java and Html languages.

The application does not possess a suitable interface: all the alterations to be effected must be done with a text editor. The applets produced will have always the same appearance if the user does not have the knowledge to modify their interface (more information about JavaTHESIS can be found in http://projects.edte.utwente.nl/pi/Java/JavaTeSystem.html).

Multiverse Esim Builder

Multiverse eSim Builder is a tool that allows us to create Web pages where the applet that represents the simulation model will be inserted. This application allows users with some knowledge of the Java language to represent models in a simple way.

Multiverse eSim builder is made of three tools that allow us to create and develop educational simulations for the Web: the Model Wizard, the Interface Builder and the Deployment Manager.

Model Wizard – is used to create the structure of the code of the application. The Model Wizard creates all the Java code that supports the interface of the application or the simulation model. In this phase the pupil or teacher define the parameters and methods to use in the model. The relation between the various parameters is done by the user using the Java language.

[1] JavaTHESIS is a product developed at the University of Twente, Netherlands.
[2] Multiverse eSim Builder is a product of Multiverse Solutions Ltd.
[3] Easy Java Simulations (Ejs) was developed at the University of Murcia, Spain.
[4] AgentSheets is a product of AgentSheets, Inc.
[5] MacTHESIS is also a product developed at the University of Twente, Netherlands.

Interface Builder - with this module the interface of the application can be created. In the interface of the model some objects, like buttons, scrollbars, images, counters, etc., can be inserted. These objects will be linked with the parameters and variables of the system.

Deployment Manager - is a tool integrated in the Interface Builder that allows the creation of the applications for the Web.

An important characteristic of MultiVerse eSim Builder is the separation of the code that defines the behaviour of the simulated model from the code that defines the visualization of the system. This characteristic is very important, because it allows to develop easily several interfaces for the same model, and to apply this same model in diverse situations. For instance, a teacher can build different interfaces for different situations, based on a specific model. These interfaces can also be easily shared by several users (Thomas and Close and McAndrew, 2000, Thomas, and Liddle and Austin, and McAndrew, 1998).

Easy Java Simulations

Easy Java Simulations (Ejs) is a tool that allows the building of simulations that represent a given model in the form of Java applets, through the visualization of the different states that this model can assume. Each one of these states is described by a set of variables.

Ejs has three main modules:

1. Introduction: in this module it is possible to introduce the model explanations or another type of information that the teacher or author intends to present to the users of the simulation. This information will appear in the beginning of the simulation Web page that will be automatically created by the tool.
2. Model: model variable declaration, initialisation and relations are specified in this module.
3. View: this module allows the definition of the simulation interface; the interface can be constituted by images, buttons, scrollbars, textboxes, etc.

To represent a model in Ejs it is necessary:

1. to declare the variables - In a first phase it is necessary to define which variables are necessary to represent the model. When adding a variable it is necessary to identify its type and initial value.
2. to represent the model itself - The second phase consists of the representation of the mathematical model of the simulation. It is in this part of the application that the user has to identify the system independent variable and the relation that exists between it and the other variables of the system. Ejs allows the user to introduce the equations of the system in two different ways: writing them in a mathematical form, being the corresponding Java code generated automatically, or directly in Java, which makes it more difficult to represent the model.
3. to show the model – The last phase is the elaboration of the graphical interface of the simulation. This is done with the module View. In Ejs it is sufficiently simple to create the interface, because the user just has to make use of a library of objects (menus, basic items, figures, etc.) to represent the variables of the system. The relation between variables and graphical objects is established accessing the object properties and indicating the variable that the object represents.

The Ejs environment is quite friendly and for simple models it is sufficiently easy to use. Ejs was created to be used by a great number of users, with or without knowledge of Java programming. However this is not true for more complex models as it is necessary to have knowledge of the language to build sound models in these cases (in http://fem.um.es/Ejs_es/index.html more information about this tool can be found).

Agentsheets

AgentSheets is a simulation tool based on agents that can be used by a very wide range of users. With this application interactive simulations and games can be created.

AgentSheets intends to be an innovative environment that combines agents, spreadsheets and the Java language in one working environment.

In AgentSheets, agents are programmable objects, which react to mouse and keyboard events. Agents can change, for instances, their appearance and position. These agents live in a world called a worksheet.

The application working environment is composed by several elements, among which the more important are: the Gallery, the Worksheet and Ristretto.

Gallery - agents in a simulation are represented in the Gallery window; to each agent one can define a specific behaviour. The management of agent behaviour is done with the aid of the programming language Visual AgenTalk (VAT), which is a language based on rules that allow characterizing agent behaviour and actions. When an agent in the gallery is created, a certain behaviour is associated with it, and it is this behaviour that will condition the action of the agent and the execution of the simulation. With the behaviours of the agents we can associate methods, actions and conditions. Actions, conditions and methods can be easily configured using the VAT language.

Worksheet – all simulations are represented in a worksheet. The size of each cell in the worksheet is determined when a new simulation is created. All worksheets possess tools for adding, removing or selecting agents and with these tools the user can define the ‘world’ of the simulation. All worksheet windows have buttons placed at the bottom to make it possible to execute, interrupt, restart or configure the speed of the simulation.

Ristretto - Through this option an applet of the simulation can be built and integrated automatically in a Web page to visualize it (for additional information about AgentSheets see the site tool at http://www.agentsheets.com).

COMPARATIVE ANALYSIS

After analysing the tools mentioned, it was possible to elaborate the Table 1 and to define a set of criteria that we consider valid to make an evaluation of these tools.

Table 1. Tools for Web-Based Simulation

	Applications			
	EasyJava	**Multiverse**	**Java Thesis**	**AgentSheets**
Application area	Several	Several	Several	Several
Model editing	In the environment	In the environment	In a text editor	In the environment
Model representation language	Java	Java	Java and Html	Java and AVT
Model representation	Graphical, simple to use	Graphical, simple to use	Text mode	Graphical, simple to use
Multimedia items	Yes	Yes	Yes	Yes
Interface	Easy to understand	Easy to understand	Does not apply	Easy
Exploration	Easy, when familiar with Java language and model equations	Easy, when familiar with Java language and model equations	Easy, when familiar with Java language and model equations	Easy
Documentation	Reasonable, some examples	Good, a few examples	Not much, some examples	Good, lots of examples
Animation	Reasonable	Reasonable	Weak	Very good and elucidative
Simulation Control	It allows interrupting, modifying simulation speed and parameters.	It allows interrupting, modifying simulation speed and parameters.	It allows interrupting, modifying simulation speed and parameters.	It allows interrupting, modifying simulation speed and parameters.
Creation of the Web page	Automatic	Through the Deployment Manager module	Automatic	Through the Ristretto application

From the above table we can see that for some criteria the tools perform well. But this is not the case for all.

All the analysed tools, except AgentSheets, are simulation tools that make use of the Java language. Although this language is being widely known nowadays, the pupil or teacher who uses these tools will have to have basic knowledge of the language to use the tools.

The specification of the model in JavaTHESIS, Multiverse eSim Builder and Easy Java Simulations is done through the direct introduction of the equations that represent the system; this introduction is done in the application environment, except for JavaTHESIS that uses a text editor. In the case of AgentSheets the representation of the model is done graphically which simplifies the task for the user. However, some models can not be represented due to the limitations of the VAT language.

The pupil or the teacher when he intends to elaborate the interface of the simulation finds in all the tools, except in JavaTHESIS (model and interface are fully developed in a text editor), a graphical environment that allows to relate the parameters and variables of the model with the objects (textbox, list, scrollbar or graphs) that will represent the former.

All the tools allow the creation of Web pages with the applet that simulates the simulation model embedded. In these applet Web pages the user can usually modify the speed of the simulation, restart or modify the parameters of the simulation.

GENERIC CARACTERISTICS FOR EDUCATIONAL WEB-BASED SIMULATION ENVIRONMENTS

Most pupils and teachers are familiar with the diverse services provided by the Web. The great success of the Web results mainly from its: navigation system, simplicity of use, motivation, reply time, democratic access, and independency.

Simulation tools for the Web will have to respect these characteristics of success, along with the following specific characteristics:

1. To be specifically designed for education with pedagogical intentions behind.
2. Simple interface - The interface will have to be simple and friendly, in order to facilitate the navigation and the use of the tool resources.
3. Intuitive to use - the great majority of simulation tools current available are not very easy to use, compel the pupil or the teacher to acquire technological knowledge to be able to benefit from these type of applications or to know the architecture and language of the application.
4. Flexible - Allow the creation of products for diverse levels, ages, application areas, based on different types of models.
5. Graphical Interface at model level - Some of these applications compel the user to introduce the relations between the elements of the model through equations, which implies that he has to have some knowledge about the model representation form.
6. Reusable - to allow the reuse of models and interfaces.
7. Resource Integration - like text, sounds, images, animations, etc.
8. Teacher/student mode - Tools that possess working environments for the teacher and the student, allow the teacher, for instances, to adapt easily simulations or models previously developed and provide a much more simple tool for the student.
9. Efficient.
10. Portable.
11. Fast.

CONCLUSION

The use of simulations in education stimulates the development of capacities like understanding, reasoning, reflection, intuition, decision-making, creativity and generalization, more and more essential in a global and changing world (Alessi and Trollip, 1985).

However simulation development, be it for the Web or not, is not an easy task as it is usually expensive, lengthy, rises frequently difficult pedagogical design questions, demands multidisciplinary teams and detailed accompaniment.

The use of dedicated tools can help surpass some of these difficulties. Yet, after reviewing some of the currently available Web-based simulation tools, we still identify some problems. Among the most important we may point:

1. The utilization of several tools still needs a substantial knowledge of programming
2. and of model representation mechanisms.
3. Some tools have poor and complicated interfaces.
4. They are limited in the types of models they permit to simulate.

REFERENCES

Alessi, S. M. and Trollip, S. R. (1985). Computer-Based Instruction, Methods and Development. Prentice-Hall, Inc.

Atolagbe, T., Hlupic, V., Taylor, S. (2000). Genisa: A Web-based interactive learning environment for teaching simulation modelling. Proceedings of the 2001 Winter Simulation Conference.

Bravo, J., Ortega, M., Verdejo, M.F. (1999). Planning in Distance Simulation Environments. Proceedings of the Comunications and Networking in Education (COMNED'99). Aulanko, Hämeenlinna. Finland.

Kuljis, J. and Paul, R. (2000). A Review of Web simulation: whither we wander? Proceedings of the 2000 Winter Simulation Conference.

Marcelino, M. J. (1998). Contributions to the Development of Educational Applications that involve Simulation Techniques. PhD Thesis. Science and Technological Faculty. University of Coimbra, Portugal.

Jong, T. and Sarti, L. (1994). Design and production of Multimedia and Simulation based Learning Material. Dorchester: Kluwer Academic Publishers.

Thomas, R.C., Close, A., and McAndrew, P. (2000). Tailoring simulations for teaching and learning - the potential of the MultiVerse Environment, International Journal of Electronic Engineering Education, vol. 32, no. 1, pp 1-12.

Thomas, R.C., Liddle, J., Austin, W.J., and McAndrew, P. (1998). Multiverse - lowering the barriers to widespread use of Web-based simulations in education. in Proceedings of the International Conference On Web-Based Modeling & Simulation, San Diego, California.

Brigas, C.
Polytechnic Institute of Guarda
6301-559 Guarda
Portugal
Email: brigas@ipg.pt

Marcelino, M. J.
Departament of Informatics Engineering
University of Coimbra
3030 Coimbra
Portugal
Email: zemar@dei.uc.pt