ELSEVIER

Simulation Practice and Theory 9 (2001) 37–54

SIMULATION PRACTICE AND THEORY

www.elsevier.com/locate/simpra

An appraisal of web-based simulation: whither we wander?

Jasna Kuljis [a,*], Ray J. Paul [a]

[a] *Department of Information Systems and Computing, Centre for Applied Simulation Modelling, Brunel University, Uxbridge, Middlesex UB8 3PH, UK*

Received 19 July 2000; received in revised form 9 March 2001

Abstract

The direction that web-based simulation modelling is taking is determined and deliberated. Environments and languages for web-based simulation are reviewed, particularly Java-based approaches. Web-based applications are discussed. After proposing a summary of the review, ways of working that will have an unpredictable effect on the future of simulation modelling are proposed. The future direction of web-based simulation is speculated on, given the argument pursued in the paper.

Keywords: Web-based simulation; Simulation languages; Java; Distributed simulation

1. Introduction

There is a growing interest in using the web as a new "platform" for applications (e.g., [17]). The simulation community has not stayed immune to the trend. The pressure imposed by the proliferation of web uses has forced the simulation community to migrate to the web in order to remain "alive". Thus, the first session on web-based simulation at the Winter Simulation Conference appeared in 1996 [9,13,40]. Since then a specific forum to bring together researchers to address the issues related to new developments on the web and simulation was held in 1998, the first conference dedicated to web-based modelling and simulation [14]. The conference covered issues concerning web agents, distributed simulation, simulation toolkits/products and applications.

* Corresponding author. Tel.: +44-1895-203-081; fax: +44-1895-251-686.
E-mail address: jasna.kuljis@brunel.ac.uk (J. Kuljis).

PII: S 0 9 2 8 - 4 8 6 9 (0 1) 0 0 0 3 2 - 5

The common theme of the 1998 conference was the recognition of the importance of Java as the *sine qua non* programming language and of the CORBA architecture. CORBA is a specification for an emerging technology known as distributed object management, which allows applications to communicate with each other no matter where they are located or who has designed them. It specifies a standard for communication between objects in a distributed client–server environment. This enables applications to transparently access the services offered by local and remote objects by the use of an object interface, independent of hardware and software implementations [41]. The trend towards Java (plus JavaBeans) and CORBA remained dominant in the subsequent conference [7,8].

However, contributors to the subsequent web-based modelling and simulation conference ran out of steam and after the year 2000 saw a big drop in papers presented the conference was discontinued. A valuable source of information is Paul Fishwick's survey on the web [15]. Another valuable web source was maintained by the MITRE web-based simulation project [39]. However, this page has not been updated since the completion of the project in January 1999. There are only a few papers published in journals. Journal of the Operational Research Society (JORS) has published a paper about a prototype Java system by Pooley and Wilcox [50]. Transactions of the Society for Computer Simulation International have published a paper on the automatic generation of simulation-based web courses (e.g., ecosystems, [2]), and Simulation produced a special issue on web-based simulation in July 1999 which was largely a mixture of very specialist application areas (uninhabited aerial vehicles, agricultural crop simulations) or pilot studies. Few journals have published future directions. The INFORMS Journal on Computing is a notable exception and in its special issue in 1998 on "Charting New Directions in OR and CS" (volume 10, number 4), it published several papers discussing web opportunities. ACM Transactions on Modeling and Computer Simulation published a panel paper [43], which, unusually for a refereed journal paper, exposes emotive opposing positions on the subject.

On reading the papers on the web and simulation, it is hard not to conjecture that a fascination with new technologies drives the trend more than the needs of industry for simulation modelling tools on the web. However, there is certainly scope for certain type of applications to utilise the advantages of Internet computing and the web. The most likely candidates are:

- applications that deal with huge quantities of data like, e.g., meteorological modelling;
- applications that enable users on multiple sites to collaborate in model design;
- applications that require direct input from a customer like, e.g., manufacturing which offers customised products;
- applications in education and training which increasingly has to cater for distance learning students.

Internet technology, in particular the web, is strongly focused on end users and collaboration. It is therefore not surprising that web-based applications raise much more interest than traditional ones. As far as web-based simulation is concerned, there are currently two main approaches, both based on the Java language. One approach develops a new simulation language using Java, and the other approach ports an existing simulation language, such as GPSS, and creates it in Java [54]. Advances in Java development tools and JavaBeans technology that can facilitate developments of reusable components in combination with the ability to distribute and execute models via the Internet may foster increased activity in the development of high-level, domain-specific simulation tools aimed at the end users. However, there is no evidence to date of this ease of portability.

Our review in web-based simulation focuses on two main areas: simulation languages and applications. In Section 2, we introduce new software technologies dominated by Java and Section 3 provides an overview of Java-based simulation environments. In Section 4, we present application domains for web-based simulation. Section 5 provides a summary of the review and our views on the state of the art of web-based simulation. In Section 6, we give some recommendations on any web-based simulation approach, which might improve its prospects. Section 7 gives conclusions and some views on the future of the web-based simulation.

2. New software technologies

Java, an object-oriented programming language, was introduced in late 1995 and quickly gained popularity due to its claimed platform independence, class reusability, and Internet capabilities. A Java component technology JavaBeans which was released with Java 1.1 in early 1997 further enhanced reusability. Components are self-contained elements of software that can be controlled dynamically and assembled to form applications. The integration of the web and Java represents a technological advancement that enables a fundamentally new approach to simulation modelling which Healy [42,43] attributes to the role the Java language plays in both the specification and implementation of the model.

Possible benefits of using JavaBeans include platform neutrality, Internet capability and visual programming. JavaBeans is a set of classes and programming conventions that constitute a component development model for the Java language. Beans are designed to be manipulated graphically within visual development environments. Skilled programmers can build and make available beans for other developers, who may be experts in their domain but may have less technical and programming expertise, to assemble visually into custom applications.

Java is now so ubiquitous that it might appear unnecessary to comment on it. For completeness the reader is reminded that [46,47]:

- simulation models in Java can be made widely available;
- an applet can be retrieved and run and does not have to be ported to a different platform or even recompiled or relinked;
- there is a high degree of dynamism because Java applets run on a browser;
- Java built-in threads make it easier to implement simulation following the process interactive paradigms;
- Java has built-in support for providing sophisticated animations;
- Java is smaller, cleaner, safer and easier to learn than C++, allegedly.

Java is the main language used in the web-based applications and it also dominates web-based simulation. Kilgore et al. [26] list Java features that have the potential to dramatically change the process used to build computer simulation models. Kilgore et al. [26] claim the following:

1. Java will became a common foundation for all simulation tools because it is the only object-oriented programming environment that effectively supports standardised components.
2. Java will foster execution speed breakthroughs through convenient and robust support for distributed processing of simulation experiments on multiple processors.
3. Java will improve the quality of simulation models as the development of application-specific software components redirects the emphasis of simulation software firms toward modelling and away from modelling development environments.
4. Java will expose the benefit of computer simulation to a larger audience of problem solvers, decision makers, and trainers since models can be distributed and executed over the Internet using standard browser software on any operating system and hardware platform.
5. Java will accelerate simulation education because students already familiar with object-oriented design, Java syntax and Java environments will no longer require instructions in specific simulation tools.

The above claims exemplify the current overwhelming enthusiasm for Java. However, time will show which of the above claims can be justified.

Displaying a web page is not any longer a simple matter of displaying text and images. With the introduction of the programming capabilities offered by Java, Javascript, and other programming languages, as well as new web standards such as XML [6], the web page is a dynamic entity. Paralleling the advances in web technology, advances in computing technology provide capabilities that were until recently beyond the reach of the machines generally available. Evidence of these advances exists in a form of a 3D interactive visualization language, the Virtual Reality Modeling Language (VRML), which has been standardized and is widely available. VRML offers opportunities for demonstrating the value of distributing 3D web content over the web.

The Extensible Markup Language (XML) is the universal format for structured documents and data on the web, and is used to express HTTP requests and responses. There are potentially many document types that can be stored as distributed simulation data. Some of these document types have widely-accepted or standardized formats. Examples of these include the many kinds of CAD files (DXF, IGES, etc.), image files (GIF, TIFF, BMP, etc), and executables (EXE, COM, BAT, DLL, etc.). However, in the domain of manufacturing simulation many documents do not have standardized format. McLean and Riddick [36] give as examples of such documents schedules, BOMs, and process plans. They quote XML as a mechanism which will allow the definition of extensible formats for new document types without adversely affecting the rest of the DMS architecture or interfaces. XML allows for definition data that has semantic information in addition to data values. XML data-type-definitions (DTD) may be used to define new document formats.

3. Environments and languages for web-based simulation

There are several Java-based discrete simulation environments. They are mostly Java versions of existing simulation languages. A number of Java-based programming tools support textual model descriptions, like simjava and Silk. Most of these support graphical views of a model and its instrumentation. More ambitious projects, like e.g. WBSE, attempt to offer complete web-based environments for model construction.

3.1. simjava

This is a process-based discrete event simulation package for building working models of complex systems. It includes facilities for representing simulation objects as animated icons on screen. simjava simulations may be incorporated as "live diagrams" into web documents. The package has been designed for simulating fairly static networks of active entities which communicate by sending passive event objects via ports and is therefore appropriate for hardware and distributive software systems modelling [19]. A simjava simulation is a collection of entities each running in its own thread [20]. These entities are connected together by ports and can communicate with each other by sending and receiving event objects. A central system class controls all the threads, advances the simulation time, and delivers the events. The progress of the simulation is recorded through trace messages produced by the entities and saved in a file.

3.2. DEVSJAVA

DEVSJAVA is an environment built in Java, which is based on Discrete Event System Specification (DEVS), object orientation and the web/Internet [51]. It supports High Level Architecture (HLA), agent-based modelling, and System Entity Structure. DEVSJAVA environment provides the foundation upon which higher constructs of DEVS (e.g., endomorphism and variable structure) can be created using the basic, as well as the Internet-based features, of Java programming. The DEVS atomic and coupled models can be developed and simulated both as standalone applications and applets. A user of DEVSJAVA is able to experiment with any DEVS model from any machine at any time and to interactively and visually control simulation execution.

3.3. JSIM

JSIM is a Java-based simulation and animation environment supporting web-based simulation as well component-based technology. By utilising component-based technology, in this case JavaBeans, the environment is built up from reusable software components that can be dynamically assembled using visual development tools [38]. In JSIM, simulation models may be built using either the event package (Event-Scheduling Paradigm) or the process package (Process-Interaction Paradigm)

[40]. JSIM supports a good graphical environment for displaying queues, and uses a Java database for storing results. The objective of the JSIM project is to provide a flexible, platform-independent, user-friendly environment for developing and running simulation models on the web. JSIM is open source software that is freely available on the web [37] and can be modified by anyone.

3.4. JavaSim

JavaSim is a Java implementation of the original C++SIM simulation toolkit at the Department of Computing Science, University of Newcastle upon Tyne, Newcastle upon Tyne, UK [33]. JavaSim is a set of Java packages for building discrete event process-based simulation.

3.5. JavaGPSS

The JavaGPSS compiler is a simulation tool which was designed for the Internet. The objective was to create a GPSS implementation which could truly be run as an applet in any Internet browser. Previous solutions developed at the University of Magdeburg had to use CGI scripts to transfer GPSS source code entered in an HTML form to a server. This server would then run the simulation with a commercial GPSS/H implementation and deliver the results back to the client. The JavaGPSS compiler is a Java program that translates GPSS source files to Java source code. Since Java applets cannot write files, the current version of JavaGPSS itself cannot be run as an applet. The output of JavaGPSS (the actual simulation) can be run in any Java-capable Internet browser [27].

3.6. Silk

Silk [18,24,25] is a commercially available general-purpose simulation language based around a process–interaction approach and implemented in Java. Silk is designed as a tool for building self-contained, reusable modelling components and domain-specific simulators. It provides a visual modelling environment where Silk-based modelling components can be graphically assembled using JavaBeans to create simulation applications in software environments such as Symantec's Visual Café, IBM's VisualAge, and Microsoft's J++.

3.7. WSE

The Web-enabled Simulation Environment (WSE) combines web technology with the use of Java and CORBA. Iazeolla and D'Ambrogio [21] claim that the WSE environment provides transparent access to simulation models and tools (location transparency, distribution transparency and platform independence); dynamic acquisition, instantiation and/or modification of simulation models; global availability (Internet-based interaction); and plug-and-use architecture to easily embed simulation models and tools.

3.8. Summary

This short review illustrates the earlier point made about most application packages being either modifications of existing packages into Java or Java for its own sake. There is a little new to report on and in fact, since many of these application packages do not support colour animation, we believe that if anything, web-based simulation is inferior to single machine packages.

4. Application domains

One of the common themes in most applications of the web is using the web as the repository of data or models. Our survey found that most applications fall in the following domains:
- Military;
- Science and Engineering;
- Education and Training;
- Manufacturing.

There are attempts to migrate simulation to the web in other domains as well. Sections 4.1–4.4 provide a short overview of applications in the above domains.

4.1. Military applications

The Internet has become a large repository of information and that potential is recognised by the US army. A web-based system called The Army Standards Repository System (ASTAR) was developed by the Army Model and Simulation Office to enhance the army's decentralised, consensus-based standards development process [35]. A web-based tool SNAP facilitates the Standards Development Process. The aim is to develop standards to improve interoperability and credibility while also increasing its commonality and reuse.

OpSim is a system designed specifically for the mission operational environment as an aid for decision makers and their staffs during ongoing operations. It is a discrete-event, aggregate-level, distributed system implemented in Java for use in a web-based environment [52]. It is designed to run in near real-time so that the progress of the real operation can be compared against the planned operation. It supports queries by external agents for state information or some hypothetical questions while it is running. OpSim allows the user a large degree of control over the simulation clock and control of the near time constraints so that the user can choose what happens if the simulation begins to lag too far behind real time.

4.2. Scientific applications

Large-scale computer simulations can take days to run and can produce massive amounts of output. Sometimes this may involve multiple hardware platforms, a variety of software applications, data stored on many devices in many formats, and little standard metadata. The exploration of simulation results can then be a laborious and highly specialised process requiring knowledge of a variety of systems. Long et al. [34] report on a web-based tool Sim Tracker which addresses problems in such environments primarily in the physics and engineering application areas. Physicists compare their many sets of results with other computational results and with experimental data, in an effort to understand how well their simulations predict physical events. A user typically runs dozens or even hundreds of variations. Managing the resulting output has been handled on an ad hoc basis by each user, mainly by manually keeping track of their computational plans and results. Their data is scattered across multiple platforms, uses a variety of formats, is difficult for them to keep organised, and is typically inaccessible to their colleagues. Sim Tracker automatically generates metadata summaries that serve as a quick overview and index to the archived results of simulations. The summaries provide convenient access to the data sets and associated analysis tools.

The Weather Scenario Generator (WSG) is intended to mine a very large array of environmental data and provide results to a user at interactive speed [23]. WSC is designed to allow a modelling and simulation customer to intelligently access environmental data for inclusion and integration with model runs. According to Kihn et al. [23] it will automate the generation of a synthetic natural environment scenario database by allowing online users to search for specific weather conditions in existing real data archives using a server side fuzzy logic engine as well as performing integrated networked searches of the Master Environmental Library (MEL) data archives. The WSG is a Java-based, networked, client–server system based on a high-performance cluster of parallel computers intended to provide an integrated and physically consistent simulation environment for an authoritative and realistic representation of atmospheric, oceanic, and/or space natural environment elements for specified regions, time frames, and conditions. The WSG is intended to allow a user to fully specify their scenario requirements via a desktop client, which then triggers the generation of environmental data sets on demand.

4.3. Education and Training

Exploration, discovery-based, and learning by doing, are valuable methods of learning which give a learner the feeling of involvement. Learning how to build models is thus best done by building models. This approach can be best facilitated if tools and models are freely and widely available and not just in the dedicated laboratories. Web-based modelling environments can be best suited for such an approach in education. The availability and interactive nature of web-based simulation also provides a good medium for students to experience the complexity and dynamism of collaborative work and can be natural environments that combine distance education, group training and real-time interaction.

A project at Heriot-Watt University, the Multiverse simulation environment [53], has been built using Java and web technology in order to facilitate the use of web-based simulations in teaching and training. Multiverse has been designed to help overcome problems actually encountered in the production of educational simulations. It facilitates the saving of classic or interesting simulation states, and linking them with supporting hypermedia to produce a "web resource base" which could be made available to other teachers in the domain. Likewise, information relating to simulation use such as frequently asked questions, assignments set for students and teachers experiences of the simulation can be saved in such a way. Kreutzer [28] reports on a similar project. Their Java-based simulator for queuing scenarios facilitates exploration of different modelling styles (e.g., event-driven, activity-driven, and process-driven; or material and machine-oriented structures). Other attempts to provide training environments on the web include a group distance exercise system implemented in Java [4,16].

Various military applications constitute the bulk of training environments migrating to the web. Blais and Serino [5] report on the migration process of computer-based training for the Marine Corps in the US. The Marine Air-Ground Task Force Tactical Warfare Simulation (MTWS), fielded in 1995, is a distributed simulation system designed to support command staff training for the United States Marine Corps.

4.4. Manufacturing

Major market trends are driving manufacturing from mass production, where the manufacturer tells the customer what to buy, to mass customisation, where the customer tells the manufacturer what to make. The Internet supports this transformation with global communication between customers and manufacturers. However, as Baker et al. [3] report, the physical realities of manufacturing impose requirements for more than just communication. They argue that manufacturing enterprises must exist over the Internet as an efficiently managed distributed enterprise. The AARIA project [3] supports the hybrid "info-mechanical" domain of a virtual manufacturing operation through three components:

- an agent architecture that decomposes the system to address both information modularity and the physical realities of manufacturing;
- a simulator that allows agents to reason about the consequences of their actions in the physical world before actually executing them; and
- an infrastructure to support the implementation of these agents.

Given the current pressures on manufacturing toward distribution, agility, and mass customisation, Baker et al. [3] see manufacturing moving to become a network of flexibly interconnected manufacturing capabilities where the Internet offers a backbone and a medium for customer interaction with the factory floor.

4.5. Other applications

Web-based simulation of autonomous software agents is another example of web-based environments used to explore the potential of the web and new software technologies. These systems are typically implemented in Java. For example, one such environment enables users to interactively change agents behaviour in a multiagent ecosystem [10], and another one explores a multiagent system for electronic commerce applications [55].

The problem of controlling crowds in public places like airports, train terminals, and sporting events is addressed by Bruzzone and Signorile [7]. They developed a prototype crowd control simulation system using Java. The model, supported by graphical components representing the environment, is intended as a tool for the operations managers that can be used over the network to control the ebb and flow of the crowd. The model also allows testing emergency procedures like fires in order to identify the efficiency of evacuation procedures.

Modelling human behaviour and the human brain are another area investigated as possible candidates for web-based simulation. Canova and Tyler [11] describe how the web can provide an environment for modelling the adaptive behaviour of humans using genetic algorithms. A repository of exemplary models that would be freely accessible on the web together with mechanisms to use such models is described by Alexander et al. [1]. It allows a modeller to create new models by using existing models and developing only those features unique to their model.

Collaboration in simulation involves the cooperation of users, who are often geographically distributed, in the execution of the same simulation. Such collaboration can include monitoring and debugging with the help from experts. An interactive simulation tool Jane is designed to help users to collaboratively interact with parallel network simulations on the Internet [48]. Jane provides mechanisms for users to incorporate their own model-specific views to override or supplement the default views and users can view their own network-specific run-time animations.

5. Summary and conclusions from the review

Our conclusion of the review into web-based simulation is that not much new is being reported. The developments are surprisingly conservative and lack the very entrepreneurial nature of the media they try to conquer. We cannot but observe that most of the work is a playground for the idle for whom it is most important what tools are being used (e.g., Java and JavaBeans) than how the new medium can enhance our use of simulation as a modelling vehicle. This is partially due to the fact that there are no real applications and no real users who are pushing for a more adventurous approach. As it is, most applications are "invented" and answer the question "What can I put on the web?" rather than "This simulation has to be available over the web, how can I design it to facilitate the needs of its users?" Applications that are not user driven rarely progress far or have an impact on society and this is the sad story of web-based simulation so far.

There is potential for simulation on the web, surely? There are many devotees and, more importantly, the US Department of Defense (DoD) is very keen to push it. So, what is wrong? In our opinion it is the mismatch between the web main characteristics and the approach taken by web-based simulation so far, which has not departed from the classical approach in order to take advantages of the new media. This mismatch is illustrated well in Table 1.

Table 1
Comparison between web features and simulation

Web features	Web-based simulation	Classical simulation
Common standards	Common standards	No common standards
Platform independence	Platform independence	Platform dependence
Interoperability	Generally not supported	Not supported
Ease of navigation	Varies	Varies
Ease of use	Difficult to use	Difficult to use
No specialist knowledge required	Specialist knowledge required	Specialist knowledge required
Not affected by change	Affected by change	Affected by change
Unstructured	Structured	Structured

The web can serve as an operating system and as a distribution channel for applications. The main characteristics of the web are: ease of navigation and use, ease of publishing content, new distribution models, enabling a network-centric computing paradigm, and enabling new intra-business applications [22].

The web is a truly democratic and non-exclusive medium as is evident by the exponential growth of Internet subscribers. The main reasons for this are that the web is easy to use, ease to navigate through, has a consistent navigational strategy (hyper links), is forgiving (one cannot make unrecoverable errors), is easy to publish on, access to it is widely available, does not depend on proprietary software, etc. Simulation is and always was a highly specialist application area (web-based or not). Simulation is not easy to use. Some simulation software does not have an obvious or consistent navigational strategy, it is often unforgiving and confusing (often hard to recover from errors), model design requires specialist knowledge of simulation as well as of knowledge of the software used, and depends on proprietary software and thus its intricacies.

The stability of the web is not affected by the addition of new sites, removal of existing sites or any changes to sites. The stability of web-based simulation, if it is truly distributed, can easily be seriously affected or even disabled, e.g., the disappearance of any site that hosts parts of modelling environments. The web is a vast collection of often unstructured information that is always available and accessible (as long as the server holding it is "alive") regardless in which format it is (text, pictures, hypermedia). However, searching the web can be very frustrating and lengthy. Simulation uses data that is not generally accepted in a standard format, usually has to be provided in a specific format, and exchange between different simulation systems is generally not supported as yet [31]. Modelling components and rules of assembling these components also have to follow standards specific for simulation software. Outputs from the simulation again follow different rules and vary in: what is reported, how it is reported, at which stage of the simulation run it is reported, formats of output files, etc.

6. Conjecture: simulation tools for the web (natural born webbers)

A large proportion of the current generation of students entering higher education, including the developed countries, are already familiar with the pastime of browsing the web and playing computer games. Browsing and adventure games encourage the participant to try out alternatives with rapid feedback, avoiding the need to analyse a problem with a view of deriving the result. Such web users, in order to use simulation, need and desire development tools that allow for fast model building and quick and easy experimentation. Furthermore, such web users should have a natural affinity to the use of simulation models as a problem understanding approach [44,45].

Web-enabled simulation analysts will be opposed to classical software engineering approaches and methodologies. They will be seeking tools that will enable them to assemble rather than build a model. This can be facilitated by providing various repositories over the net. The web can provide an easy access to various repositories which can hold modelling components, past simulation models, real data archives, etc. These repositories can then be used selectively. For example, model development can be facilitated by selecting the appropriate repository and then selecting from it model components which are relevant to the model builder's modelling requirements. According to Whitman et al. [54] the benefits of web-based simulation may be realised by small companies, who could provide simulation model repositories over the Internet. Medium to large manufacturing companies may benefit from the distributed nature over an internal Intranet or between supply chain members across the Internet.

However, providing numerous repositories and tools on the web does not necessarily make the modellers task any easier. There is a consensus among web users that searching the web for information is often quite a frustrating task. The rapidly growing web industry is addressing this problem by constantly launching new or improved search engines and facilitating better filtering of information. However, good search engines are at their best if one has a stable starting point, e.g., a web page with all links to the relevant sites with some description of what the site contains. But in the dynamic environment of the web the most common problem is, even if this is provided, keeping the starting place up to date (i.e., deleting non-existing links, reflecting changes to the web addresses, adding new sources, etc.). Therefore, searching for appropriate tools and repositories should be supported somehow. We will call this supporting facility an environment for web-based simulation. A list of recommendations for simulation environments given by Kuljis [32] can almost entirely be applied to web-based simulation environments. These recommendations, modified to reflect web specifics, state that simulation environment should provide model developers with the following:

1. *A facility to design and/or choose a problem domain*. The structural context in the model should be expressed clearly, so that the graphical representation makes visible the relations in the model like the dependency or non-dependency of activities or events. Therefore, when defining a new problem domain the user should be able to select suitable graphical representations of the domain entities either from a repository of icons available on the system itself or somewhere on the web. If the user cannot find suitable icons a tool should be provided to either modify existing icons or imported icons or to draw new icons if necessary.

2. *A support for data input/model specification*. Some of the features which would facilitate improved data input/model specification were already reported on in [30] and include: data independence, modeless dialogue, facilities for representation of complex data structures, data validation facilities, on-line help facility, and facilities to accept data from some of the major database and spreadsheet software. Since the web is basically a huge repository of data the recommendations are particularly important and the major prerequisite for a success of any web-based environment/software. Such systems, empowered with sophisticated data import and data handling facility, would enable the selection of data from multiple web sources, in various formats, and the customising of data of interest for use by a particular model or application. Of course, we cannot ignore a growing interest in data mining and we envisage that web-based simulation systems will also require data mining capabilities.

3. *A support for visual simulation*. The graphical representation of constructs for different applications should give definite information about the type of model component it represents, such as waiting queues, customers or servers in queuing systems or stores or suppliers in store keeping systems. To enable "good" design for animation, the tools should provide a greater number of drawing object templates and an extensive colour palette and colour aiding facility; support modification of graphic objects (erasing parts of graphics, filling whole or parts of graphic objects with colours or patterns, resizing, rotating, flipping, etc.); provide on-line help; combining several graphical objects into one; and so on.

4. *A support for simulation statistics/results*. In the case where a simulation system is developed for an end-user or, in the case of educational and training systems, for a learner, there is a need to provide an explanation of the simulation results. Regardless of how attractively these results are presented, end-users often lack the mathematical background necessary for understanding the simulation results. Every simulation system, especially one developed for the end-user, should enable the display of simulation results independently from the processing. This means that the results can be examined after or during the simulation run. The sequencing of the results display should be left to the user hence providing the user with greater flexibility in using the system. The modeller should have a facility after the simulation experiment, or if dynamic graphs are used during the simulation experiment, to modify the graph type and scaling to an appropriate form for the actual data.

5. *User support and assistance*. If interactive on-line tutorials are available as an option within help screens, much ambiguity and many answers to what-if questions would be resolved. It is easy to integrate animation in these tutorials as well and, thus, provide full power that tutorials can offer. Multiple styles of user interfaces can be supported by careful design of an application's functional operations and by customising the user interface to the needs of end users in each class.

Although the above recommendations are general and therefore not made specifically for web-based simulation environments these recommendations still apply with some modifications related to the nature and the source of model components.

7. Conclusions

7.1. The immediate future

We have not covered some issues commonly discussed under the general umbrella of web-based simulation like, e.g., parallel distributed simulation and the HLA. The former in our opinion has not much relevance to the developments on the web per se. The HLA is certainly talked a lot about as the mandated standard for distributed simulation by the US DoD. The HLA Baseline Definition was completed and approved as the standard technical architecture for all DoD simulations in 1996 [12] and in December 1997 was accepted as a draft IEEE standard to be supported by the Simulation Interoperability Standards Organization. The HLA is implemented as a part of the DoD Modeling and Simulation Master Plan which calls for the creation of a technical framework to foster interoperability across multiple simulations, reuse simulation components, facilitate insertion of technological advances, and provide flexibility to changing user requirements.

The US DoD is certainly the major proponent of migration of simulation to the web and of enforcing standards. However, it is too early to say whether that is a sufficient reason to push simulation developers in other sectors into web uncertainty. The major requirement for this to happen is not yet fulfilled, i.e., the eagerness of simulation users to do so. As we already commented the developments so far are not user driven and until that happens, in our humble opinion, web-based simulation will continue to be just a playground for eager aficionados of the web.

7.2. Whither we wander?

It is now possible to develop environments with coherent web-based support for collaborative model development, dynamic multimedia-based documentation, as well as open widespread execution and investigative analysis of models. The ability to access multimedia on the web clearly introduces greater potential for the use of videos of problem scenarios, and for interaction with stakeholders situated at remote locations. For example, when the running model hits an unknown combination of circumstances, an expert stakeholder might be able to determine the successful rules for advance. Today society is increasingly relying on video cameras to increase security and safety of our environment. Video cameras are ubiquitous and cover activities on streets, public places, in shops, in banks, on roads, factories, etc. For example, a Hong Kong container terminal has a computer-controlled television control centre which has 100% video coverage of the terminal. Whilst its purpose is clearly for security and safety, it requires little imagination to visualise how a simulation of the terminal operation could call up the appropriate video camera when problem discussants get to the point of a simulation run where clarification is desired. Similarly, design of a new road system can use a simulation model which uses video footages of traffic on indicative road points in various time slots.

Some have predicted that the software industry will be divided into component factories where powerful and general components will be built and tested, and into component assembly shops where these components will be assembled into flexible business solutions. Such component-based development, if it occurs, might give significant gains in productivity and quality as well as known obvious benefits to web-based software development. Web-based simulation can easily subscribe to such an approach. A review of component-based simulation is given by Pidd et al. [49]. In a way, lot of current simulation modelling environments, in particular visual simulation modelling environments, look like component factories where models are assembled from various components.

The question of portability remains unresolved. Components can be recompiled upon delivery so this would be no problem. Java remains more of a promise than a realisation. The new approach to simulation described above by "natural born webbers" might require the innovative developers to solve the problem of portability as well.

In conclusion, we have shown that the web and its users are different to "one-person-on-one-machine" computing. However, most web-based simulation is not only an attempt to transport the latter on to the web, but with less proficiency than at present available. We anticipate that a new breed of web-literate users will produce innovative web-simulation approaches. We wait for such paradigm-shifting activity [29] with eager anticipation.

References

[1] A. Alexander, M. Arbid, A. Weitzenfeld, Web simulation of brain models, in: A.G. Bruzzone, A. Uhrmacher, E.H. Page (Eds.), Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation, The Society for Computer Simulation International, San Francisco, 1999, pp. 29–33.

[2] M. Alfonseca, J. deLara, Automatic generation of simulation-based web courses and model documentation, Transactions of the Society for Computer Simulation International 17 (3) (2000) 106–119.

[3] A.D. Baker, H.V.D. Parunak, K. Erol, Agents and Internet: infrastructure for mass customization, IEEE Internet Computing 3 (5) (1999) 62–69.

[4] E. Berglund, H. Eriksson, Distributed interactive simulation for group-distance exercise on the web, in: P.A. Fishwick, D.R.C. Hill, R. Smith (Eds.), Proceedings of the 1988 International Conference on Web-Based Modeling & Simulation, The Society for Computer Simulation International, San Diego, CA, 1998, pp. 91–95.

[5] C.L. Blais, S.L. Serino, Adaptation of a wargaming system for web-based operation, in: A.G. Bruzzone, A. Uhrmacher, E.H. Page (Eds.), Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation, The Society for Computer Simulation International, San Francisco, CA, 1999, pp. 61–68.

[6] T. Bray, J. Paoli, C.M. Sperberg-McQueen, Extensible markup language (XML) 1.0 specification, October 6, 2000. http://www.w3.org/TR/REC-xml, Accessed: 4th of March 2001.

[7] A.G. Bruzzone, R. Signorille, Crowd control simulation in Java based environment, in: A.G. Bruzzone, A. Uhrmacher, E.H. Page (Eds.), Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation, The Society for Computer Simulation International, San Francisco, CA, 1999, pp. 23–28.

[8] A.G. Bruzzone, A. Uhrmacher, E.H. Page (Eds.), Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation, The Society for Computer Simulation International, San Francisco, CA, 1999.

[9] A.H. Buss, K.A. Stork, Discrete event simulation on the World Wide Web using Java, in: Proceedings of the 1996 Winter Simulation Conference, Coronado, CA, 1996, pp. 780–785.
[10] A.M.C. Campos, D.R.C. Hill, Web-based simulation of agent behaviour, in: P.A. Fishwick, D.R.C. Hill, R. Smith (Eds.), Proceedings of the 1988 International Conference on Web-Based Modeling & Simulation, The Society for Computer Simulation International, San Diego, CA, 1998, pp. 9–14.
[11] B.S. Canova, J.G. Tyler, An adaptive distributed architecture for near real-time genetic algorithm execution, in: P.A. Fishwick, D.R.C. Hill, R. Smith (Eds.), Proceedings of the 1988 International Conference on Web-Based Modeling & Simulation, The Society for Computer Simulation International, San Diego, CA, 1998, pp. 96–101.
[12] Department of Defense, USA, DMSO, http://www.dmso.mil/, Accessed: March 4, 2001.
[13] P.A. Fishwick, Web-based simulation: some personal observations, in: Proceedings of the 1996 Winter Simulation Conference, Coronado, CA, 1996, pp. 772–779.
[14] P.A. Fishwick, D.R.C. Hill, R. Smith (Eds.), Proceedings of the 1988 International Conference on Web-Based Modeling & Simulation, The Society for Computer Simulation International, San Diego, CA, 1998.
[15] P.A. Fishwick, Web-based simulation, http://www.cise.ufl.edu/~fishwick/websim.html, Accessed: March 4, 2001.
[16] R. Granlund, Web-based micro-world simulation for emergency management training, in: A.G., A. Uhrmacher, E.H. Page (Eds.), Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation, The Society for Computer Simulation International, San Francisco, CA, 1999, pp. 47–52.
[17] A. Guru, P. Savory, R. Williams, A web-based interface for storing and executing simulation models, in: J.A. Joines, R.R. Barton, K. Kang, P.A. Fishwick (Eds.), Proceedings of the 2000 Winter Simulation Conference, Orlando, FL, 2000, pp. 1810–1814.
[18] K.J. Healy, R.A. Kilgore, Silk: a Java based process simulation language, in: Proceedings of the 1997 Winter Simulation Conference, Atlanta, GA, 1997, pp. 475–482.
[19] F. Howell, R. McNab, simjava: a discrete event simulation library for Java, in: P.A. Fishwick, D.R.C. Hill, R. Smith (Eds.), Proceedings of the 1988 International Conference on Web-Based Modeling & Simulation, The Society for Computer Simulation International, San Diego, CA, 1998, pp. 51–56.
[20] F. Howell, R. McNab, simjava, http://www.dcs.ed.ac.uk/home/hase/simjava/, Accessed: January 29, 2000, but not accessible on March 4, 2001.
[21] G. Iazeolla, A. D'Ambrogio, A Web-based environment for the reuse of simulation models, in: P.A. Fishwick, D.R.C. Hill, R. Smith (Eds.), Proceedings of the 1988 International Conference on Web-Based Modeling & Simulation, The Society for Computer Simulation International, San Diego, CA, 1998, pp. 37–42.
[22] R. Kalakota, A.B. Whinston, Electronic Commerce. A Manager's Guide, Addison-Wesley, Reading, MA, 1997.
[23] E.A. Kihn, M.N. Zhizhin, S.J. Lowe, A.V. Troussov, R.E. Englebretson, R.A. Siquig, The weather scenario generator, in: A.G. Bruzzone, A. Uhrmacher, E.H. Page (Eds.), Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation, The Society for Computer Simulation International, San Francisco, CA, 1999, pp. 233–241.
[24] R.A. Kilgore, E. Burke, Object-oriented simulation of distributed systems using Java, and Silk, in: J.A. Joines, R.R. Barton, K. Kang, P.A. Fishwick (Eds.), Proceedings of the 2000 Winter Simulation Conference, Orlando, FL, 2000, pp. 1802–1909.
[25] R.A. Kilgore, K. Healy, Java, enterprise simulation and the Silk simulation language, in: P.A. Fishwick, D.R.C. Hill, R. Smith (Eds.), Proceedings of the 1988 International Conference on Web-Based Modeling & Simulation, The Society for Computer Simulation International, San Diego, CA, 1998, pp. 195–201.
[26] R.A. Kilgore, K.J. Healy, G.B. Kleindorfer, The future of Java-based simulation, in: Proceedings of the 1998 Winter Simulation Conference, Washington DC, 1998, pp. 1707–1712.
[27] U. Klein, S. Straßburger, J. Beikirch, Distributed Simulation with JavaGPSS based on the High Level Architecture, in: Proceedings of the 1988 International Conference on Web-Based Modeling & Simulation, The Society for Computer Simulation International, San Diego, CA, 1998, pp. 85–90.

[28] W. Kreutzer, Teaching queuing scenarios on the world-wide-web, in: A.G. Bruzzone, A. Uhrmacher, E.H. Page (Eds.), Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation, The Society for Computer Simulation International, San Francisco, CA, 1999, pp. 41–46.
[29] T.S. Kuhn, The Structure of Scientific Revolution, University of Chicago Press, Chicago, 1962.
[30] J. Kuljis, User interfaces and simulation models, Journal of Simulation Practice and Theory 1 (5) (1994) 207–221.
[31] J. Kuljis, HCI and simulation packages, in: J.M. Charnes, D.J. Morrice, D.T. Brunner, J.J. Swain (Eds.), Proceedings of the 1996 Winter Simulation Conference, Coronado, CA, 1996, pp. 687–694.
[32] J. Kuljis, The usability of user interfaces to simulation systems, in: R. Macredie, R.J. Paul, D. Anketell, B. Lehaney, S. Warwick (Eds.), Modelling for Added Value, Springer, London, 1998, pp. 215–238.
[33] M.C. Little, JavaSim home page, http://javasim.ncl.ac.uk/, Accessed: March 4, 2001.
[34] J. Long, P. Spencer, R. Springmeyer, SimTracker Using the web to track computer simulation results, in: A.G. Bruzzone, A. Uhrmacher, E.H. Page (Eds.), Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation, The Society for Computer Simulation International, San Francisco, CA, 1999, pp. 171–176.
[35] L.E. McGlynn, LTC.D. Timian, Army model and simulation standards – from producer to consumer; How the process works, in: A.G. Bruzzone, A. Uhrmacher, E.H. Page (Eds.), Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation, The Society for Computer Simulation International, San Francisco, CA, 1999, pp. 75–80.
[36] C. McLean, F. Riddick,The IMS mission architecture for distributed manufacturing simulation, in: J.A. Joines, R.R. Barton, K. Kang, P.A. Fishwick (Eds.), Proceedings of the 2000 Winter Simulation Conference, Orlando, FL, 2000, pp. 1539–1548.
[37] J.A. Miller, JSIM: A Java-based simulation and animation environment, http://orion.cs.uga.edu:5080/~jam/home/, Accessed: March 4, 2001.
[38] J.A. Miller, Y. Ge, J. Tao, Component-based simulation environments: Jsim as a case study using Java Beans, in: D.J. Medeiros, E.F. Watson, J.S. Carson, M.S. Manivannan (Eds.), Proceedings of the 1998 Winter Simulation Conference, Washington DC, 1998, pp. 373–381.
[39] A Survey of Web Based Simulation, MITRE web based simulation project, http://ms.ie.org/websim/survey/survey.html, (1999), Accessed: March 4, 2001.
[40] R.S. Nair, J.A. Miller, Z. Zhang, Java-based query driven simulation environment, in: Proceedings of the 1996 Winter Simulation Conference, Coronado, CA, 1996, pp. 786–793.
[41] Object Management Group, The Common Object Request Broker: Architecture and Specification, 1995.
[42] E.H. Page, A. Buss, P.A. Fishwich, K.J. Healy, R.E. Nance, R.J. Paul, Panel session: the modeling methodological impacts of web-based simulation, in: Proceedings of the 1998 International Conference on Web-Based Modeling & Simulation, San Diego, CA, 1998, pp. 123–128.
[43] E.H. Page, A. Buss, P.A. Fishwick, K.J. Healy, R.E. Nance, R.J. Paul, Web-based simulation: revolution or evolution?, ACM Transactions on Modeling and Computer Simulation 10 (1) (2000) 3–17.
[44] R.J. Paul, D.W. Balmer, Simulation Modelling, Chartwell-Bratt Student Text Series, Lund, Sweeden1993.
[45] R.J. Paul, V. Hlupic, The CASM environment revisited, in: J.D. Tew, S. Manivannan, D.A. Sadowski, A.F. Seila (Eds.), Proceedings of the 1994 Winter Simulation Conference, Orlando, FL, ACM, New York, 1994, pp. 641–648.
[46] R.J. Paul, Web-based simulation: Whither we wander? A contribution in the Panel session: the modeling methodological impacts of web-based simulation, in: Proceedings of the 1998 International Conference on Web-Based Modeling & Simulation, San Diego, CA, 1998, pp. 123–128.
[47] R.J. Paul, Web-based simulation: Whither we wander? in: E.H. Page, A. Buss, P.A. Fishwick, K.J. Healy, R.E. Nance, R.J. Paul, Web-Based Simulation: Revolution or Evolution? ACM Transactions on Modeling and Computer Simulation 10 (1) (2000) 3–17.
[48] K.S. Perumalla, R.M. Fujimoto, Jane – an architecture for interactive parallel simulation, in: A.G. Bruzzone, A. Uhrmacher, E.H. Page (Eds.), Proceedings of the 1999 International Conference on

Web-Based Modeling and Simulation, The Society for Computer Simulation International, San Francisco, CA, 1999, pp. 83–88.

[49] M. Pidd, N. Oses, R.J. Brooks, Component-based simulation on the web? in: P.A. Farrington, H.B. Nembhard, D.T. Sturrock, G.W. Evans (Eds.), Proceedings of the 1999 Winter Simulation Conference, Phoenix, AZ, 1999, pp. 1438–1444.

[50] R. Pooley, P. Wilcox, Distributing decision making using Java simulation across the World Wide Web, Journal of the Operational Research Society 51 (4) (2000) 395–404.

[51] H.S. Sarjoughian, B.P. Zeigler, DEVSJAVA: Basis for a DEVS-based collaborative M&S environments, in: P.A. Fishwick, D.R.C. Hill, R. Smith (Eds.), Proceedings of the 1988 International Conference on Web-Based Modeling & Simulation, The Society for Computer Simulation International, San Diego, CA, 1998, pp. 29–35.

[52] J.R. Surdu, G.D. Haines, U.W. Pooch, OpSim: a purpose-built distributed simulation for the mission operaional environment, in: A.G. Bruzzone, A. Uhrmacher, E.H. Page (Eds.), Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation, The Society for Computer Simulation International, San Francisco, CA, 1999, pp. 69–74.

[53] R.C. Thomas, J. Liddle, W.J. Austin, P. McAndrew, Multiverse – lowering the barriers to widespread use of web-based simulations in education, in: P.A. Fishwick, D.R.C. Hill, R. Smith (Eds.), Proceedings of the 1998 International Conference on Web-Based Modeling & Simulation, The Society for Computer Simulation International, San Diego, CA, 1998, pp. 110–113.

[54] L. Whitman, B. Huff, S. Palaniswamy, Commercial simulation over the web, in: Proceedings of the 1998 Winter Simulation Conference, Washington DC, 1998, pp. 335–339.

[55] M. Yoo, W. Merlat, J. Briot, Modeling and validation of mobile agents on the web, in: P.A. Fishwick, D.R.C. Hill, R. Smith (Eds.), Proceedings of the 1988 International Conference on Web-Based Modeling & Simulation, The Society for Computer Simulation International, San Diego, CA, 1998, pp. 23–28.