Simulation Practice and Theory, vol. 9, no. 1-2, October 2001, pp. 55-72

Research and Commercial Opportunities in Web-Based Simulation

John A. Miller, University of Georgia

Paul A. Fishwick, University of Florida

Simon J.E. Taylor, Brunel University

Perakath Benjamin, KBSI, Inc.

Boleslaw Szymanski, Rensselaer Polytechnic Institute

Keywords: Web-Based Simulation, Components, Agents, Cooperative Work, Commercialization.

Abstract

In a few short years, Web-Based Simulation, has exhibited explosive growth in the simulation research community. This paper which grew out of a panel discussion at the 2000 WEBSIM Conference, briefly reviews the development of this research area and considers future opportunities, both on the research and commercial sides. On the research side, the area is maturing, but still quite active. Efforts to expanded Web-Based Simulation to include new capabilities beyond those found in conventional simulation technology or provide interoperation with other information processing technology are particularly promising. On the commercial side, a critical mass of research knowledge is now available. However, some catalyst is needed to produce any substantial movement of simulation vendors toward Web-Based Simulation. This could happen quickly under a couple of scenarios: A small simulation vendor focussed on Web-Based Simulation could begin to claim significant market share. Alternatively, the development of a "killer-app" to demonstrate a clear advantage to Web-Based Simulation could make this shift happen very quickly.

1 Introduction

Web-Based Simulation began shortly after Web browsers became readily available. Early efforts began in 1995, first by providing Web front-ends to simulations running as Common Gateway Interface (CGI) scripts/programs. In addition, work began on Java based simulation packages, systems and environments that would run anywhere on the Web. Results from these activities were first reported in simulation conferences starting in 1996 (*e.g.*, there was one session [7, 2, 10] on Web-Based Simulation in the 1996 Winter Simulation Conference). From these beginnings, the interest and level of research effort increased very rapidly. In 1998, an entire conference dedicated to Web-Based Modeling and Simulation (WEBSIM) began which continues on today.

Now that Web-Based Simulation has been around for about half a decade, it is a good time to take inventory of how far the field has come, but more importantly where it is going. At the 2000 Web-Based Modeling and Simulation Conference, a panel was formed to address this very issue. This article is detailed report on the findings of this panel.

The panel was moderated by Dr. Miller and the panelists consisted of Drs. Fishwick, Benjamin, Szymanski and Taylor. The rest of this article is organized as follows: In section 2, Dr. Fishwick will present his perspective on future research directions for Web-Based Simulation. In section 3, Dr. Benjamin will discuss some fundamental programming paradigms that are essential for continued progress in Web-Based Simulation (*e.g.*, object, component and agent). At a higher level, appropriate software architectures must be considered (*e.g.*, Multi-Agent Systems (MAS), Distributed Artificial Intelligence (DIA) and High-Level Architecture (HLA)). This discussion will lead into section 4, where Dr. Szymanski will consider software architectural as well as implementational issues for Web-Based Simulation applied to large-scale simulation and multi-modeling projects. Finally, Dr. Taylor will address the issue of when and how Web-Based Simulation will impact the commercial side of simulation. Responses from other panelist will also be included in each section. A brief summary will highlight some of the significant discussion.

2 Research Directions (Dr. Fishwick)

Now that several prototype Web-based simulation systems and environments have been developed and studied, future work should build on top these successes. Ultimately, this work will likely broaden the notion of simulation and widen its applicability. In this section, a perspective on future research in Web-based simulation is offered.

1. *What areas/concepts fit under the umbrella of Web-Based Simulation (e.g., distribution, componentry, interoperability, embedding, visualization/animation, VR)? Which of these areas need further work?*

   Web-Based Simulation (WBS) is a term used to capture a cross-section of simulation and Web based research and technology. If we categorize simulation into model design, model execution and execution analysis then we are prone to couple the Web with each of these components to see what offspring are created. To see the connection between these two areas, take an arbitrary concept or tool in simulation and then apply the question "how would this be made Web-ready?" In modeling, we are concerned with the design and structure of objects which, in their connections, provide behavioral and dynamic information about the phenomena of study. A bond graph models the flow of energy and the Petri net, the flow of tokens. In non-Web modeling enterprises, models are either drawn on paper or an interface is created whereby the model is drawn on the computer via a graphical user interface (GUI). A Web modeling approach offers the possibility of drawing inside the browser through Javascript or Java. More dramatic modeling forms can use additional forms of representation, from a low-level declarative XML (Extensible Markup Language) file format to a 3D VRML (Virtual Reality Modeling Language) world, whose 3D components represent model components. Beyond this surface difference of GUI on the desktop versus GUI inside the browser, the more interesting aspect of Web-based modeling involves group participation in the modeling process. The Web brings to bear the idea of collaboration and connectivity, whereas the browser is just one vehicle for supporting this. Not only can the automobile be designed and modeled in a virtual, collaborative space, but the bond graph capturing the thermodynamics of the engine can likewise be shared. This sort of sharing opens up all kinds of possibilities. Model execution and analysis also play significant roles in WBS, and I have not discussed them in depth only because significant work in Java-based simulation applications has been done and is described elsewhere in this paper, whereas the marriage of the Web and the process of modeling remains a nascent technology. Very little work has been done applying Web-based analysis and output visualization within the simulation community, but this suggests future research. Moreover, a tight coupling between modeling, execution and analysis is enhanced in a Web-based environment since groups associated with each simulation activity are brought closer together. All in all, it is this opportunity for sharing simulation data, models and knowledge that I find more interesting than how to distribute or code the execution in a particular Web-oriented language such as Java.

All areas need further work with an eye toward keeping up in the race to place everything on the Web. This is true for Simulation as it is for every other area. In WBS, we have done some considerable work in Java-based applications. The work is impressive, and has resulted in several Java-based simulation packages. This work should continue and prosper. However, I feel that far too little work has been done in modeling and analysis. If you look at the papers in WBS, they are mostly targeted at a relationship between the Java language and the general problem of programming a simulation. While this is a natural relationship, we need to start expanding our horizons into model design and analysis. How will models be shared over the Internet? What metaphors can be used and what model representation languages can be fostered? With the emergence of XML, and its related technologies, we need to consider whether XML can serve as an adequate representation language for expressing the structure of models and simulation data.

Response from Dr. Taylor.

Many areas that fall under the umbrella of WBS have been indicated and some interesting research directions have been discussed. One may make an observation concerning what one may call the "Race for the Web." In the wildly expanding area of e-commerce, for example, there is a gold-rush pioneer spirit that has driven the development of many, many web-based systems. These have met with varying degrees of success (or failure). Simulation has not yet joined the Race. This may work in the favor of WBS in two ways. First, the web is a moving target. XML is to replace HTML. New linking schemes are replacing old. Java changes. The tools that we have to hand to create WBS are changing constantly. Secondly, there does not appear to be the critical mass of interest from the simulation community to even begin the Race. The consequence of this? We have time to reflect and consider the fundamental issues of WBS and attempt to avoid the pitfalls that e-commerce has encountered.

What are the fundamental issues of WBS? Before answering this consider the lack of simulationists that have the cross cultural skills required to appreciate both the subtle requirements of the simulation modeler and the technological infrastructure of web-based applications. Simulationists tend to fall into two categories, those who have much experience in simulation modeling methodology, but no idea of the potential of the web, and those who have knowledge of simulation modeling infrastructure (the technologies that one may use to support the process of simulation modeling) but little idea of the needs and requirements of simulation modeling practitioners. For example, one well versed in infrastructural issues may suggest that virtual reality is a useful technique in simulation. VRML, for example, gives us the opportunity to to develop sophisticated virtual environments that one may interact with. Its use? It may be suggested that VRML could augment/replace the animation environments used in commercial simulation software. If this is the case, what could it add to the simulation modeling process? How important is visualization for problem definition, conceptual modeling, experimentation, verification, validation, etc.? These are questions that a technologist could not answer without significant consultation with a practitioner of simulation. The consequence of this is that

WBS could suffer from a plethora of white elephants; technological wonders that have little or no use to the simulation modeler. In conclusion, there are many areas that fall under the umbrella of WBS and there is much work to be done. The main approach that should be taken to carry out this work must begin by bringing together the mainly disparate methodology and infrastructure communities by a program of education. Only by accomplishing this can we be certain of laying a firm foundation on which WBS can be built.

2. *Can Web-Based Simulation stimulate a paradigm shift in simulation (e.g., model-theoretic, information-theoretic, a facet of some larger concept)?*

   Yes, it can be seen as a paradigm shift. The shift is one from a simulationist hovering over his computer, running experiments and analyzing results to one of global proportions with multiple interacting researchers and practitioners. These people are sharing knowledge. The Internet started this revolution, but the Web put clothing on the Internet. Traditionally, academic simulationists have not stressed human-computer interaction. We have a long way to go in this area since the Web technology suddenly opens up new doorways to our neighbors.

   Response from Dr. Taylor.

   An important aspect of this is accessibility. A major impact of WBS means that simulations, and the information that a simulation generates, can be more readily accessed than is currently the case. WBS, however, provides a wider context for simulation to be used. How often does one require all the facilities of a simulation application? Consider the situation where a simulationist had created a series of model runs where each animated model view illustrated, for example, certain potential problems with the production line under study. For the simulation project to continue, the simulationist requires that each manager studies the animated views and reports back on their observations. In this case all that is needed is the animated view, not the whole simulation package. The world wide web provides an extremely suitable mechanism for applications such as this. To further expand the comments made above, if the shift to WBS can be made, then it will play an important role in information-based fields such as knowledge management.

3. *How can knowledge from the domains of cooperation (e.g., business process modeling and workflow) and collaboration (e.g., CSCW) be applied in Web-Based Simulation?*

   CSCW is an area which should explode within the simulation community. It really hasn't been touched, when compared with the preponderance of Java-based manuscripts, if one counts papers in proceedings and journals. Most simulationists think in terms of an individual sitting at a desk, creating a model. The problem with this scenario is that it is somewhat archaic and inefficient. WBS leads to the possibility of model libraries or repositories situated over the Web, in companies, universities and government institutions. When it is time to advance decision making through simulation, the complex model needs to be constructed from Web-based modeling components. Creating a monolithic model from scratch, or by one person, will not provide enough leverage for using the technique of simulation in the first place. Simulation is expensive, but the Web is going to make it cheaper. Reuse of model components will drastically reduce the time needed for model development. It is nice to be able to create Java components, but these do not take the place of higher-level, and more visual, model components that enable workers to more clearly understand the system under study.

Response from Dr. Benjamin.

The emerging practice (in the business process modeling and workflow communities) of building reusable process and workflow templates and model libraries needs is useful for Web-Based simulation. Distributed templates and libraries of simulation model components increase the cost effectiveness of modeling and analysis. Recent advances in distributed collaboration technologies may be applied advantageously for Web-Based simulation. An example of this is the use of distributed simulation modeling for Virtual Prototyping (VP) and Simulation Based Design (SBD). VP and SBD are used extensively in the aerospace, automotive, and shipbuilding industries.

3 Components and Agents (Dr. Benjamin)

Because Web-based software development involves programming-in-the-large, it is important to use state-of-the-art techniques from software engineering, distributed systems and artificial intelligence. Currently, there has been substantial progress in the application of component-based and agent-based approaches. In this section, three questions are posed and addressed to highlight issues related to Web-based simulation.

1. *Is there agreement on the definitions of (a) object, (b) component and (c) agent. How are they similar and how are they different?*

   Although there is no universal agreement on the definition of the terms - objects, components, and agents, these technologies play an important role in enhancing the effectiveness of modeling and simulation. An object is a structured collection of information. Objects are characterized by the properties they exhibit, the relationships they bear with one another, and the behaviors that they are capable of. Object behavior is manifested by the flow of messages between objects. The object-oriented approach to software development includes additional concepts such as encapsulation, inheritance, and polymorphism, and has been successfully used for software development over the past decade.

A component is an independently deployable set of software services. The component-based approach to software development is an area of intense research and development in the information technology community. This approach seeks to provide the mechanisms that will allow developers to assemble software systems from collections of re-usable parts or building blocks. This approach has been very successful in the hardware world as evidenced by the "plug and play" hardware peripherals for computers. The path to success has been considerably slower in the software arena owing to inherent complexity of software. The convergence to a relatively robust set of "standard" languages and protocols in the past few years augurs well for increasing plug compatibility in the software domain. Prominent technologies that facilitate component-based software development include COM, DCOM, ActiveX, IIOP, RMI, Java Beans, and CORBA. Related techniques are design patterns, analysis patterns, and object oriented design and analysis methods. Software components and the component based approach can be used to facilitate both distributed simulation model development and simulation model application on the Internet. The approach suggested by MOOSE [4] discusses elements of the former. The use of components for deploying simulations to enable distributed analysis has been the focus of many recent initiatives including JSIM [9, 15] and IDES [12].

Software agent technology is an emerging area that appears to offer significant benefits to the simulation community. The concept of an agent was first introduced by Carl Hewitt in his concurrent actor model. An agent is defined as a software entity which functions continuously and autonomously in a particular environment, often inhabited by other agents and processes. Key characteristics that agents should exhibit include: (i) Autonomy: where agents have internal states and goals, and they behave in a manner that enables accomplishment of goals on behalf of its user, (ii) Cooperation: where agents must be able to cooperate with other agents or human users, and (iii) Learning: where agents must improve their performance over time. Agents that exhibit learning characteristics are called intelligent agents.

The primary motivation for component technology is re-use. Components address some of the technical and pragmatic barriers to re-use that object technology failed to address. Components have been relatively more successful facilitating cost effective software development, deployment, and evolution. The characteristics of autonomy and cooperation distinguish agents from both components and objects. The capacity to learn adds another level of sophistication that components and objects lack.

Response from Dr. Miller.

Objects provide an ideal foundation for reusable component technology. From one point of view, if dynamic interaction between objects is supported, then you have components. Typically, an object will need to be able to find, load and dynamically call methods of other objects (introspection and reflection capabilities). It is also useful to be able to save/restore the state of components (persistence and serialization capabilities). Greater flexibility in component interaction is achieved if an event capability is supported. Finally, components are even more useful if they can interact remotely (distribution) or universally (platform independent/Web-based). One could build an agent based system on top of component technology. If distributed components has enough flexibility, self sufficiency and *autonomy* to provide high-level services to other agents or users, they may be viewed as agents. This also requires flexible, high-level communication between agents to convey semantics (e.g., using KQML or something equivalent, say in XML) to facilitate *cooperative* collaboration. Some agents may embed sophisticated decision making or learning capabilities, making them more *intelligent* than typical agents.

2. *When will reusable and composable software components become widely available and effectively usable in simulation? What still needs to happen to make this a reality? It has been a goal for a long time.*

   Software components and the component based approach can be used to facilitate both distributed simulation model development and simulation model application on the Internet. The approach suggested by MOOSE [4] discusses elements of the former. The use of components for deploying simulations to enable distributed analysis has been the focus of many recent initiatives including JSIM [9, 15] and IDES [12].

   The technical challenges that limit the use of component technology for simulation modeling are the same barriers that limit its use within the larger software community. These technical barriers include (i) trade-off between composability and utility, (ii) knowledge sharing barriers (shared ontologies and semantic interoperability), (iii) lack of standards, and (iv) legacy systems. The benefits of using components for simulation modeling and analysis include (i) more effective global resource exploitation through distributed modeling and analysis, (ii) increased modeling agility and robustness, (iii) lower simulation life cycle costs, and (iv) enhanced knowledge management (capture, maintenance, and re-use of knowledge).

Response from Dr. Miller.

From a Web-based simulation point-of-view, an effective and easy-to-use platform independent component technology is needed. Although, component-based development has been a goal for a long-time, many elements are now coming together to bring componentry into wide-use. Time will tell if Java with its client-side components (Java Beans), server-side components (Enterprise Java Beans) and distributed components (Jini) will combine sufficient power, ease-of-use and efficiency to usher in the new era. Come to think of it, that is a tall order.

3. *Should the commercial sector adopt HLA or would it be better off with a yet to be developed HLA-Lite as suggested by Mike Pidd?*

   The High Level Architecture (HLA) community has successfully leveraged object and component technology to develop useful tools for simulation modeling and analysis for military applications. It appears that the commercial simulation community can benefit from the experiences and insights gained by HLA researchers and practitioners. Although preliminary efforts have been to transition HLA concepts and tools to other communities much work remains to be done. The HLA-Lite idea proposed by Mike Pidd is a step in the right direction.

Response from Dr. Taylor.

The observation that the HLA specification, and the run time infrastructures that have been developed, have been successful is true from a technological perspective. The DMSO advocates the HLA as a general purpose architecture for simulation reuse and interoperability. The HLA has also been adopted as the Facility for Distributed Simulation Systems by the Object Management Group and is in the process of becoming an IEEE open standard.

If this is the case then HLA-Lite is possibly not the right direction in which to move as this could detract from an emerging standard rather than add. Instead the rules, interface specifications and templates of the HLA should be reconsidered from the perspective of the large investment made by the commercial (non-military) sector in simulation modeling software. The need is there. It would be difficult to argue the need for a distributed, real-time simulation for training as most simulation models are used for analysis and planning (and therefore need to run very much faster than real-time). However, a clear motivation is reuse. Consider a company that has a simulation of a production line and a simulation of a warehouse. It may be the case (and often is) that each of these simulations is created using a different simulation package (Witness, Taylor, etc.). Most of the formats used by these packages are incompatible (a model saved by one cannot be opened by another). Currently, if one were to attempt to link the two simulations together the only recourse would be to recode one simulation in the other's package. This may be prohibitive in terms of cost and time. This is also the case with supply chain management where relevant simulations may not only be written in different packages, but may also be proprietary (one company may wish to do business with another but may not be willing to share all of its secrets!) There are many other issues to be considered in addition to simply linking simulations together (the semantic content of models, for example). However, if it were possible to reuse the simulation models by linking them via some form of connective middleware (such as the HLA), then a wide range of attractive possibilities could be opened up to commercial simulation.

4 Modeling-in-the Large (Dr. Szymanski)

To make modeling-in-the-large over the Web a reality, issues beyond components and agents must be addressed. These issues include concepts and techniques for multi-modeling and meta-modeling as well as interoperability and scalability issues.

1. *What is the relationship between Web-Based Simulation and Multi/Meta-Modeling?*

   Meta-modeling involves linking different models of the same or related phenomena to create a fuller and/or more precise meta-model. The component models may involve different physical scales and/or different aspects of the modeled process. An example of a latter could be a plasma model representing particles and their self-induced or externally induced electromagnetical field linked with a model of chemical reactions between particles. An example of a multi-scale model is a Lyme disease simulation representing a regional spatial scale as a lattice of nodes with each node using a local model of disease dynamics.

   Often, the linked models are developed independently and run on different sites. The main concern of meta-modeling is to introduce proper interfaces, protocols as well as linkage and synchronization mechanisms to enable creation of a meta-model. The Web is a very suitable environment for developing meta-simulations thanks to the Web's wide scope and constant availability. Ideally, no redevelopment, shipping of codes or new computational resources should be needed to join together independently developed and run models. In addition to computational resources, the model expertise can also be distributed to the development teams that built component models, fostering debugging and verification of the created meta-model. An ease of announcing availability of models and of participation in meta-modeling over the Web is an important factor in growing popularity of meta-modeling.

Response from Dr. Fishwick.

Multi-modeling and meta-modeling are different in many respects. I have seen meta-models most often referred to in the context of, say, using a response surface instead of an underlying equational model. The equational model, when executed, creates and input-output relationship. This relationship can be used in lieu of the model, and so it creates a "meta-model." Multi-modeling as defined in [5, 6] is the process of creating a multi-layer model where, for a model that is of a specific class or type X, a component of this model can be refined or sub-defined into another model of type Y. Multi-modeling is purely structural where one model type (say a Petri net) can be refining a state of a Finite State Machine. That is, when one "zooms" into the state, one finds the Petri net underneath. Meta-modeling is also very useful, but is different in that the high level model has a non-structural relationship to the level below it. The meta-model is obtained by executing the model at the next lower level. As to the relationship between multi-modeling and the Web, the Web supports heterogeneity in model structure by virtue of the distribution of data and knowledge, which are spread over the globe.

2. *The Web is beginning to provide several massive digital libraries. Can it also provide massive model repositories? What issues does this entail (e.g., searchability, interoperability/linking, synchronization, scalability)?*

   Although the Web is beginning to provide several massive digital libraries, there is still too little standardization in today's simulation models to provide massive model repositories. In addition to standardization of models and their interfaces, there are also issues of computational resources needed to run models and synchronization between events or simulation times in component models. Current research in meta-modeling focuses on interoperability, linking and synchronization issues. As our experience with building meta-models over the Web increases, some standard interfaces and synchronization methods are likely to emerge. The component models can be linked either by low level communication primitives (e.g., TCP/IP session) or using mobile agents. An agent based approach lends itself to much more general and flexible linkages than those that can be supported by the low level primitives. For example, using mobile agents, different models can be linked together at run time. Also, the interfaces can be made transparent to each implementation by including necessary data transformation in the agent's code. Although agents introduce larger communication delays than low level implementation, they are capable of minimizing the amount of data that needs to be shipped by processing some of the data locally. An important open question in this area today is how to support efficiency of communication and synchronization in agent linked models in Web-based simulations.

In general, using the Web either as a source of inexpensive computing power or as a repository of component models requires research on efficient linking and synchronization of models over the Web.

Response from Dr. Fishwick.

One area that heralds a change for the better is XML. A key problem in model representation, and the use of models to build large repositories, is that each simulation code relies on its own model representation. In many cases, this representation is in the code itself, and sometimes when a pictorial approach is used in modeling, the representation will be a graph. Whereas HTML was a page description and markup language, XML allows us to capture the semantic tags needed for model representation. XML is not a panacea since no doubt its employment will yield a multiplicity of XML-based formats. But at least, all model formats will be in XML and so it levels the playing field so that the same basic schema language is used. This is a good beginning.

3. *What is the relationship between research in PADS, HLA, DIA and Web-Based Simulation.*

Web-based simulations are by definition parallel and distributed, yet compared to the typical PADS environment, the communication delays are an order of magnitude larger and also more unpredictable. Hence, the Web-based simulation must develop different synchronization techniques than those used in parallel simulations. Meta-modeling using Web-based modeling will benefit from any standardization of models, so availability of the models compliant to HLA or some other standard, will increase opportunities and ease of linking them. Note, table 1 defines the acronyms used in this discussion.

An interesting alternative is to develop an interface language, similar to CORBA, which will standardize available access to data and synchronization primitives of a simulation. A simulation program with such an interface becomes available for linking with other programs

Acronym	**Definition**
PADS	Parallel and Distributed Simulation
HLA	High Level Architecture
DIA	Distributed Artificial Intelligence
XML	eXtensible Markup Language
CORBA	Common Object Request Broker Architecture
IDL	Interface Definition Language

Table 1: Definitions of Acronyms

via mobile agents. It seems that XML or CORBA IDL might be too general for this purpose and a subset of a IDL is sufficient. Simplicity of such a language is important, because its compiler must capture not only data formats (which is relatively easy) but also a semantics of synchronization involved in the distributed execution. The requirement of an interface in one model to enable linking with other models seems less stringent and easier to accomplish than enforcing HLA implementation of all models to ensure model compatibility in distributed execution.

Response from Dr. Fishwick.

We need to come back to the definition of ”simulation”. If we refer to model execution, then it is natural to think of distributed simulation execution. If we refer to simulation in a broader sense, including modeling and analysis, then the execution need not be distributed.

4. *Describe some large-scale applications for Web-Based Simulation.*

An example when such linkage of models over the Web is beneficial is a sea port simulation in which management of ship movements and anchoring can be best described by a discrete event simulation system. However, at different scale of details, the acceleration and velocity of each ship is affected by the weather and water currents which are best described by the atmospheric and shallow water continuous subsystems [17]. Another example, that we recently implemented [3], is the spread of Lyme disease, which involves interactions of many species with large differences in their size and spatial and temporal scales of development (e.g., deers, mice and ticks). Another applications of these techniques include air traffic control, highway management systems, population dynamics of complex ecosystems, etc. Meta-modeling techniques are important in creating detailed models of complex, multi-scale systems and in linking existing component models into an integrated meta-model of more complex phenomena.

Response from Dr. Benjamin.

Most web-based simulation applications to date either are research and development prototypes or are still under development. [14] have developed a 3D web-based viewer called JView for visualizing distributed HLA-compliant simulation models. A prototype Parallel Discrete Event Simulation System (PDES) has been developed to facilitate the distributed analysis and optimization of simulation models on the Internet [18]. A simulation model of spacecraft processing and launch activities is currently being developed as part of a NASA project [11]. This model will be deployed for distributed viewing and analysis on the Internet. This model will be used as a decision support tool for future space initiatives.

5 Commercialization Issues (Dr. Taylor)

Although Web-based simulation is new, it is not too early to think about commercialization issues. New technological advances are typically slow to appear in commercial systems (parallel simulation technology is a case in point). Because of the ever-increasing importance of the Web, there may be pressure for faster adoption. This section discusses issues of inertia and resistance, as well as, possible catalysts for more rapid adoption.

1. *From a commercial perspective, what is Web-Based Simulation and what can it do? Why should simulation vendors or their customers be interested?*

   What is a commercial perspective? What is meant by this is the perspective taken by the majority of simulation modelling consultants and customers that use the many commercial simulation modelling packages currently employed in civil or non-military areas such as manufacturing, health, and transportation to support planning and analysis techniques such business process re-engineering and supply chain management (WITNESS, ARENA, SIMUL8, TAYLOR II, SIMPLE++, and PROMODEL are some examples). What could WBS offer to stimulate the interest of vendors or customers (simulation consultants and their customers)? Let us consider some possibilities.

(a) **Cooperative Environments.**

Simulation modelling is a fundamentally cooperative endeavour. We (the consultants and customers) cooperate when the problem/system the simulation project is to address is defined, we cooperate during the development of the conceptual and computer models, and we cooperate during experimentation and reporting. Most importantly we cooperate during the verification and validation of the models created during the simulation project. Why? It is vitally important for customers to understand what is happening during the development of a model. For example [13] have studied some factors that play a key role in the success of a simulation project. After interviewing providers of simulation expertise (simulation consultants) and their customers they observed a major factor related to success was Communication and Interaction (frequency, clarity and appropriateness of communication and interaction with those involved in the simulation project). The web provides the opportunity to develop cooperative software, i.e. the same simulation environment is loaded into each web browser and is augmented by facilities that allow each user to view/share the work of another. This may be further extended to allow users to talk to each other, to transfer files, to document cooperative sessions, etc. Each of these facilities have the added attraction of supporting remote cooperation and therefore may accrue reduced project costs as the need for travel is also reduced. One may therefore take the view that a vendor developing a WBS that supports cooperation in the simulation project may expect better sales, as those who use it may expect better communication and interaction (and possibly a better reputation for successful simulation projects) and reduced project time and cost due to the reduction in travel.

In a similar vein, WBS gives the possibility for on-line training or user support. There are many times in the life of a simulation environment when users (the simulation consultants) are faced with time consuming phone calls or visits to training centres. Many vendors now demand that any users purchasing their software attend an obligatory training course aimed at eliminating modelling problems that frequently occur. Some also provide support via email and newsgroups, as well as providing searchable databases of previously encountered problems. The web could provide an integrated solution to some of these problems, especially if a cooperative WBS environment could be used.

(b) **"Plug and Play" Environments.**

A "Plug and Play" environment refers to the trend followed by many current web browsers. When a web browser is first set up it possess the basic functionality to display and interpret a variety of basic web languages, instructions and images. The implication of this to the structure of simulation modelling software is that it may be treated in the same way. Components of the software could be purchased as and when necessary. One can envision a basic framework that can be loaded into a web browser to, for example, execute a simple process-based model. Other components could then be purchased to develop conceptual models, perform experimentation, statistical analysis, optimisation of various types, and to accommodate cooperation. It could also be that different modelling conceptual frameworks could be loaded to suit the system being studied. In other words, adopting a web-based approach to simulation software provision could lead to a new form of extensible simulation environment.

The benefits of this approach? There are several. The user need only purchase those simulation components that he or she requires at the time. If one was then also to cost software by use rather than by license (i.e. by on-line metered use) the user need only pay for the time that the software was actually used. For the vendor, there is the opportunity to develop different, more attractive cost models of their software. No vendor would intentionally lose money by moving from one sales base to another. However, the ability to develop flexible payment schemes for particular users (i.e. cheaper simulation software leading to more sales over time) may make a vendors software more attractive. Additionally, the ability to keep users up to date by making the latest version of a simulation environment component available via download may also lead to reduced vendor costs as previous versions may be phased out over time.

(c) **Model Databases and Model Reuse.**

Another attractive feature of WBS is the model database and the concept of reuse. Consider a vendor who develops a database (library) of models. A model is not necessarily a complete model of a car factory, or a port, or an airport. Indeed the validity of these models must be questioned. Is it possible to have a generic model of a particular set of systems; are all car factories so similar? The answer to this issue is irrelevant. Even if it was possible to have an "off the shelf" model of a system, the use of this model would mean that much of the communication and interaction necessary for the success of a simulation project would be lost. It is highly important for the customer of a project to be involved in the development of a model, otherwise how could validation be practised? A possibly better view is to continue to adopt the practices found in many modelling packages that are currently used. These have a set of model components (queues, workstations, resources, bays, conveyors, etc.) that are used in the development of a model. These contain features that are generic to the system component that they represent. These features are usually capable of being adapted as necessary or may be switched on or off as appropriate. The point is that they are at a level of complexity and understanding that is appropriate to the system component being modelled. A web-based model database that is accessible to WBS based on current simulation practise means that vendors can make new model components available and downloadable when necessary. This feature would be both attractive to users as new model components could be made available, and to vendors to enhance their service to the user and to present another opportunity to charge the user.

Response from Dr. Benjamin.

Web-Based Simulation is the use of the Internet and the World Wide Web to facilitate simulation modeling and analysis. Conceptually, the benefits of the web for simulation derive from the use of the web to facilitate cooperation, collaboration, and distributed work. The scope of WBS spans the life cycle of simulation modeling - problem definition, conceptual model design, input data collection and analysis, detailed model design, model verification and validation, design of simulation experiments, model execution, output analysis, model optimization, and model deployment. Another important application of distributed simulation (part of model deployment) that has been surprisingly ignored is simulation-based training. Early uses of simulation with the web have mainly focused on model validation and model execution. The potential benefits of WBS have largely been unrealized from a commercial perspective. Areas that could benefit through WBS in the near term are conceptual model design and simulation-based training. From a vendor perspective, a promising focus area is the development of libraries of re-usable and extendible simulation components and models that can be cost effectively inserted and used in web-based application environments.

2. *Since the Web-Based approach typically excludes proprietary approaches, will vendors be reluctant to adopt this approach?*

Under the previous discussion some consideration was given to the opportunities for profit under WBS. The reason for this is that in a very pragmatic sense, vendors must have some incentive to move from their current software base to a web-based one. Discussions of a new web-based world of simulation are always intriguing and intellectually stimulating. Unfortunately, in the very competitive world of simulation modelling software any move to WBS must be justified in terms of cost. Additionally, simulation software vendors have much expertise in the development of software for particular computing systems. Moving to WBS would involve a different (and costly) set of skills (i.e. developers used to creating software for a windows-based operating system cannot simply recode the software for the web). It is entirely possible that the cost incurred by a vendor in translating their software from the current operating platform into a web-based one could not be justified by a predicted increase in their future sales. Finally, and most tellingly, one can easily debate a future in which all vendors sell extensible cooperative web-based simulation environments that use models taken from shared model databases. However, discussions with leading simulation software houses all point to the same conclusion. This future will not happen until one software house takes the first step, and this will not happen until predicted revenue will more than balance out the cost of redevelopment.

Response from Dr. Benjamin.

The web-based approach to simulation may sometimes exclude proprietary approaches but this is not the main issue. The main issue as rightly noted in the previous paragraph is the cost versus benefit of migrating to web-based simulation software. Unless there is a compelling real world problem that is solved only through a web-based approach, or a significant benefit of converting to web-based solutions, the commercialization of WBS technology will be slow. There needs to be a business case for the change to WBS, that is, the cost of WBS conversion needs to be justified with compelling and sustaining revenue gains from WBS. Non commercial funding for R&D organizations to collaborate with commercial simulation vendors will likely accelerate the adoption and growth of WBS in industry.

3. *What might be the initial door openers for commercial adoption of Web-Based Simulation (e.g., collaborative modeling environments)?*

The previous discussion painted a rather bleak future for web-based simulation. Let us take the opportunity to investigate how a first step could be taken. As previously observed, the first step involves a rather cold calculation based on predicted revenue. This is be based on predicted demand, the predicted use of WBS by the users, i.e. the simulation consultants and their customers. Given that few software vendors appear (over the time of writing) to be seriously considering a move into web-based simulation, it appears that their users are not demanding a change. Why is this?

One reason why this is the case could be that users are just not interested in WBS. However, given that the field is little more than four years old and is principally academic in nature, users may not be aware of the possible benefits of demanding a move to WBS. It may therefore be a case of a lack of awareness. However, even if every effort was made to provide more education via seminars/papers at seminal conferences such as the Winter Simulation Conference or WEBSIM and Special Issues such as this one, for example, these would still be based on speculation. Possibly what is needed is more demonstrable benefits. Is it possible to give real examples of cost savings to the user via this technology?

Let revisit the observation made in [13]. After reviewing many comments made by writers in the field of simulation and after interviewing representatives of the simulation consultant and customer communities (the users) they concluded that communication and interaction were one of the keys to the success of a simulation project. Picking up on a point made earlier in this paper, a web-based *cooperative* simulation modelling environment is feasible. Can we shed more light on what is meant by this?

Computer Supported Cooperative Work (CSCW) is a multidisciplinary field that appeared in the early 1980's as a response to the failure of Information Technology to properly support the needs of people working in groups and organisations ([8]). The field gave rise to groupware, software that supports cooperative working. Groupware includes integrated multimedia document management, workflow (automatic processing and routing of media), e-mail, conferencing and calendaring applications that are in wide use in industry today. Groupware is not yet made a great impact in the world of simulation modelling. This is not due to the lack of software nor it is due to inadequate technological support (as evidenced by the existence of groupware software such as Lotus Notes/Domino and Microsoft Office/Exchange and NetMeeting, and the development of distributed/networked simulation modelling technologies). Indeed another panel of simulation experts highlighted the need for cooperative simulation modelling tools that can be used within an enterprise or between many enterprises engaged in business to business commerce to support applications such as Entity Resource Planning ([1]).

The conclusion to this? It appears that it is possible to develop approaches to the integration of groupware with simulation modelling to promote remote cooperation. If this is possible, then the benefits (cost and time savings) to the users of simulation modelling software can be demonstrated. For example, 1 shows the combination of the net-conferencing tool Microsoft NetMeeting with a simulation modelling package (see [16] for more details). If this can be done, then the benefits of WBS can be demonstrated to the user community. If this can be done, then the demand for WBS can be demonstrated to the vendor community. Finally, if this can be done, then we may have commercial WBS.

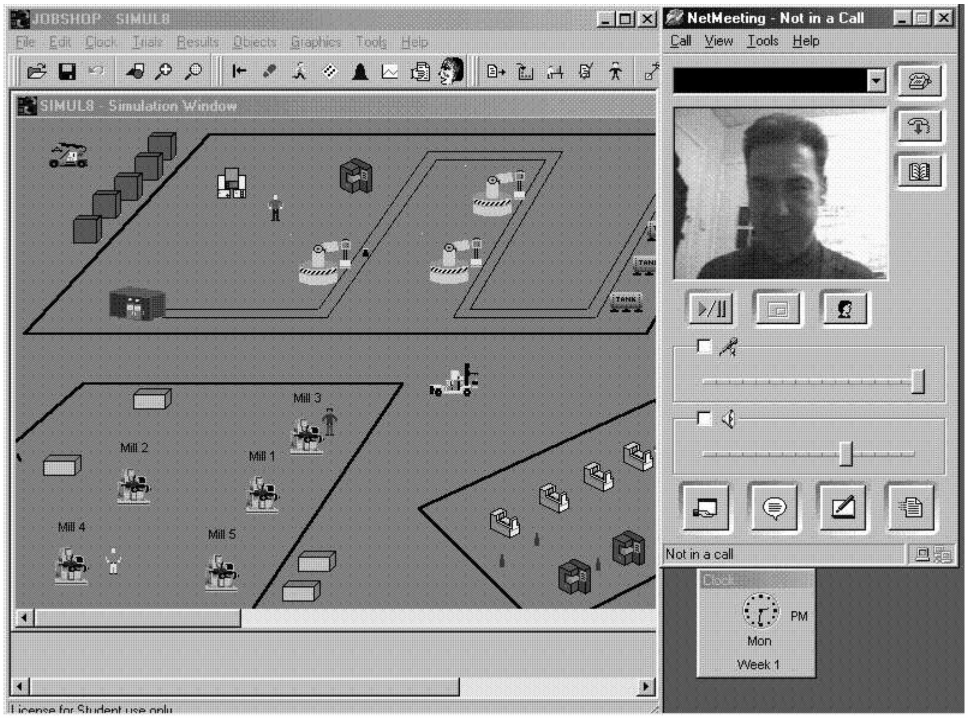
JOBSHOP - SIMUL8
File Edit Clock Trials Results Objects Graphics Tools Help
SIMUL8 - Simulation Window
TANK
TANK
Mill 3
Mill 2
Mill 1
Mill 4
Mill 5
License for Student use only
NetMeeting - Not in a Call
Call View Tools Help
Not in a call
Clock
PM
Mon
Week 1

Figure 1: Example of Tool Integration

Response from Dr. Fishwick.

The example area ”collaborative modeling environments” is certainly a key area for us. We are already beginning to see several commercial products for collaborative CAD/CAM environments where one engineer can posit a design and then another across the world can interact with the first engineer with a common 3D environment. In simulation, we have a need for an immersive 3D environment for what we are modeling. Moreover, we have a need for an immersive environment for the model structure as well. The interactive HCI relationship between the phenomena being modeled and the model structure needs lots of work. The Blaxxun Community Server is an example of a successful commercial product that allows multiple avatars sharing the same VRML world, with opportunities to interact within the world. We need more environments like this, with the important capabilities supporting shared object manipulations.

4. *Where do you see Web-Based Simulation in five years?*

   Given the previous discussion in this section, I see this field having a very steep hill to climb. Looking into the proverbial crystal ball I see two futures. In the first I see this community, not the vendors, not the consultants nor their customers, but the *research* community, in five years time producing many more academic papers on WBS and another Panel paper by another academic making similar conclusions. In the second, this community develops and keeps one foot firmly in the commercial sector and adopts an attitude of demonstration rather than promise with a view to stimulating demand in the user community. If this happens then in five years time we shall be debating the finer points of the world wide web standards with the World Wide Web Consortium that should used by the emerging commercial WBS environments that are being written. It is this future that is by far the most desirable of the two. This is the focus of the UK EPSRC Technology Programme Network GROUPSIM (see www.brunel.ac.uk/ csstsjt for more details).

   Response from Dr. Fishwick.

   I agree with cooperation with the W3C (World Wide Web Consortium). We need to better iron out directions for research involving XML and how our models will fit within XML representations for all of the domain-specific areas. All of these areas are working independently. How will our models be structured so that the representations can fit together? Given a manufacturing plant XML representation, how will we connect this to our discrete event models?

6 Summary

From the above discussions, it is apparent that the question is not "will" Web-Based Simulation become important, but rather "when" and "by what path" will it become important. For the simulation research community, there are ever-growing research opportunities as discussed. On the commercial side, vendors who see the importance of the Web and interoperation with other information technology tools, have, at least in our opinion, great future opportunities.

References

[1] Banks, J. (1998). The Future of Simulation Software: A Panel Discussion. *Proceedings of the 1998 Winter Simulation Conference*, Washington, DC, 797-804.

[2] Buss, A. and Stork, K. (1996). Discrete Event Simulation and World-Wide Web Using Java. *Proceedings of the 1996 Winter Simulation Conference*, Coronado, CA, 780-785.

[3] Chen, G. and Szymanski, B. (2000). Multi-paradigm Simulations in Modeling Spread of Lyme Disease. *Proceedings of the 2000 European Multi-Simulation Conference*, Ghent, Belgium, 631-639.

[4] Cubert, R.M., Goktekin T. and Fishwick, P.A. (1997). MOOSE: architecture of an object-oriented multimodeling simulation system. *Proceedings of Enabling Technology for Simulation Science*, Part of SPIE AeroSense '97 Conference, Orlando, FL, 22-24.

[5] Fishwick, P.A. and Zeigler, B.P. (1992). A Multimodel Methodology for Qualitative Model Engineering. *ACM Transactions on Modeling and Computer Simulation*, 2(1): 52-81.

[6] Fishwick, P.A. (1995). A Taxonomy for Simulation Modeling Based on Programming Language Principles. *IIE Transactions on IE Research*, 30: 811-820.

[7] Fishwick, P.A. (1996). Web-Based Simulation: Some Personal Observations. *Proceedings of the 1996 Winter Simulation Conference*, Coronado, CA, 772-779.

[8] Grudin, J. (1994). Computer Supported Cooperative Work: History and Focus. *IEEE Computer*, 2(5): 19-26.

[9] Miller, J.A., Ge, Y. and Tao J. (1998). Component-Based Simulation Environments: JSIM as a Case Study using Java Beans. *Proceedings of the 1998 Winter Simulation Conference*, Washington, DC, 373-381.

[10] Nair, R. and Miller, J.A. (1996). A Java-Based Query Driven Simulation Environment. *Proceedings of the 1996 Winter Simulation Conference*, Coronado, CA, 786-793.

[11] NASA (1999). Change Management and Analysis Tool. *Research and Technology 1999 Annual Report*, NASA John F. Kennedy Space Center.

[12] Nicol, D., Johnson, M., Goldsby, M. and Yoshimura, A. (1997). Development of the IDES Parallel Simulation Framework. Proceedings of the 1997 Winter Simulation Conference, Atlanta, GA, 93-99.

[13] Robinson, S. and Pidd, M. (1998). Provider and customer expectations of successful simulation projects. *Journal of the Operational Research Society*, 49(3), 200-209.

[14] Salisbury, C.F., Farr, S.D., and Moore, J.A. (1999). Web-Based Simulation Visualization Using Java3D. *Proceedings of the 1999 Winter Simulation Conference*, Phoenix, AZ, 1425-1429.

[15] Seila, A.F. and Miller, J.A. (1999). Scenario Management in Web-Based Simulation. *Proceedings of the 1999 Winter Simulation Conference*, Phoenix, AZ, 1430-1437.

[16] Taylor, S.J.E. (2000). Groupware and the Simulation Consultant. *Proceedings of the 2000 Winter Simulation Conference*, Orlando, FL, 83-89.

[17] Wilson, L., Cybenko, G., Lynch, D., Cushman-Roisin, B. and Szymanski, B. (2000). KDI: Next-Generation Agent-Based Distributed Simulation. *http://www-nml.dartmouth.edu/KDI.*

[18] Yucesan, E., Chen, C.H. and Lee, I. (1998). Web-Based Simulation Experiments. *Proceedings of the 1998 Winter Simulation Conference*, Washington DC, 1649-1654.