Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

SIMULATION APPLICATION SERVICE PROVIDING (SIM-ASP)

Thomas Wiedemann

Technical University of Berlin
Franklin Str. 28/29 FR 5-5
10587 Berlin, GERMANY

ABSTRACT

This paper considers advantages and actual problems of web based simulation systems. Based on a review of environments and languages for web based simulation, some fundamental requirements for successful simulations are presented. The goal is to change the technology-driven approach in a more customer oriented philosophy. The common concept of "Application Service Providing (ASP)" should be understood as a future technology for web based simulation environments. The paper presents a real Simulation ASP. The system is based on a database for all model and simulation data. Different simulation tools are permanently linked to the ASP control system and provide the end-user with a highly interactive work. Three different levels of modeling (source code, fixed models and universal simulation components) lead to high flexibility for all areas of simulation.

1 INTRODUCTION

The main architectures of web based simulations have been shown and tested successfully during the last years (Fishwick 1996, Fishwick et al. 1998, Healy and Kilgore 1997). The number of real applications and efficient tools for web based simulation is still very small. A critical analysis of web based simulation environments was made in (Kuljis and Paul 2000). After a review of actual web based simulation technologies, main problems are identified:

- A lot of web based simulation projects were done as a test scenario. The requirements of real customers were not taken into account.
- Often the terminus "web based" was understood only as a synonym for the ease of use, simple user interfaces and high flexibility. This is a fatal misunderstanding, because the web does not automatically solve all the old problems of simulation technology.

In conclusion, the further development of web based simulation should depend on a more user oriented approach and on a solution of old restrictions and problems of simulation technology. Only if real customers will see real advantages in speed, cost and flexibility, web based simulation will find a valuable position in future business information systems.

2 COMPARISON OF WEB TECHNOLOGIES AND WEB BASED SIMULATION

A comparison between traditional and web based simulation was given in (Kuljis and Paul 2000). The problem is, that web based simulation is just at an early stage and is not completely finished. A better starting point could be a comparison between common web technologies and traditional simulation environments.

Table 1: Web and Simulation Characteristics

	Common Web technologies	**Traditional simulation technologies**
Common standards	yes (HTML, XML, TCP/IP)	no common standard for model & results
Data handling	by unique URL	proprietary
Information Structures	tree structures and lists	very complex
Specialist knowledge	no (only clicking and reading)	yes (from a lot of scientific areas)
Navigation	easy	difficult
Ease of use	very good	very difficult
type of operations	read & analyze information	**synthesize** models and analyze results

The comparison reveals large differences concerning all criteria. A simple transformation of actual simulation techniques into the Web will result in the same disadvantages plus some new restrictions and problems caused by the web itself. The main cause for all differences consist in the **different types of operation**. A typical internet surfer almost works in a **passive, assimilating mode**. His creative work is limited to decisions about selecting the next

interesting link or new combinations of search criteria. A wrong decision can be promptly corrected by pressing the "Back"-button.

Modeling, simulation and result analysis require **a high level of active and creative work**. Almost all actions are depending on each other. The future result of many decisions is not obvious. In nearly all cases wrong decisions in model design lead to wrong results of the simulation. In contrast to the web, there are no warning messages concerning critical results or wrong model conditions. In conclusion, it seems to be very dangerous to make a extrapolation from the ease of use of common web based systems to web based simulation systems. The typical work and necessary features are very different.

A second problem of web based simulation systems is the high complexity of their user interfaces. Modern simulation systems like Taylor, Arena or AutoMod contain some dozens of very different forms and reports. The forms are connected by logical relationships. The user works in a highly interactive way and uses data-driven pull-down fields for a high comfort of modeling. The actual state of web technologies allows such forms only with large HTML tags or Java applets. But the efforts in developing a complex web based system of forms and reports are sometimes higher than the realization with state of the art development systems like Delphi or Visual C++.

Instead the question "How can we transfer existing simulation systems to web ?" we should answer the question : "What are efficient web based simulation services for real customers at the actual time ?"

3 USER BENEFITS OF WEB BASED SIMULATION

Nearly all different requirements of simulation users can be transformed into basic terms of **profit** and **time**.

The profit is calculated as the difference between development costs of a simulation study and the expected revenues from the study. Development costs are influenced by the cost of the simulation environment and the modeling philosophy and comfort. Unfortunately the starting investments are on a high level between $5000 and $50,000 for typical simulation environments or external consultants. The revenue of a simulation study is unknown in the beginning. The risk of loosing money rises with increasing investment costs. Web based information technologies allow new business models of using simulation services. Instead of a high starting investment in software, simulation services can be rent for a interval of time.

The second term "Time" is important in real decision scenarios. Often a decision must be made under special circumstances like disturbance or external, unpredictable factors. The **"time to decision**", this means the time for finding a solution, is limited to some minutes or hours. This requires a very powerful simulation environment in terms of performance and optimization capabilities. More than one license and computer could be necessary. After the decision process is done, this environment runs idle until the next decision occurs. By using modern web technologies such free simulation capabilities can be used by other decision makers. The costs for simulation studies can be reduced significantly, if the peaks of required simulation power are averaged by a common pool of simulation resources, which are shared between different customers.

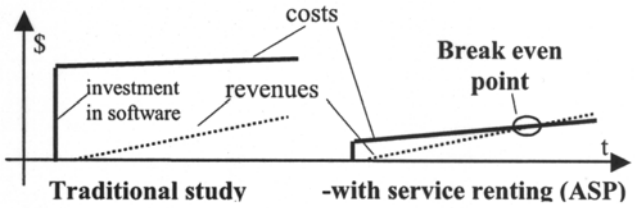

Figure 1: Costs of Traditional and ASP-Simulation

"Portability" and **"interoperability**" are often called the most important benefits of web based simulation. This is particularly correct for the current state of different computer environments. Depending on the existing hardware there are only two options - the simulation study is impossible due to incompatible technologies or the systems allow data exchange and control. For the customer, portability plays the role of a "killer" question. If this question is answered positively, costs and time for decision making again become the most important benefits of web based simulation for the customer.

4 SIMULATION AREAS WITH HIGH EFFICIENCY OF WEB BASED SYSTEMS

The costs of developing complex simulation models are always very high. Although model generators and highly sophisticated modeling techniques could be used, the efforts for basic system research, data acquisition, model verification and validation actually are still connected with human resources. As it was mentioned before, the web supports operations with information distributing characteristics very well. Tasks with a high degree of creative work, resulting in synthesis of new web objects are still executed with external programs like HTML-editors or layout programs. This means - the internal structure and available functions of the current web are not ready for a real creative developing process! This implies the following conclusions:

- The web can not simplify the modeling process in the near future. Simulation models should be developed by using traditional simulation tools..
- The web can support the reuse of existing simulation models very well by its distributed nature and the content management function like search machines and common data access protocols.

If we combine these conclusions, web based simulation systems will be of high efficiency, if the models remain nearly unchanged and only the simulation control variables and the input data are changed. Models of this kind are based on real systems with a fixed structure and dynamic working conditions, like

- flexible manufacturing cells with N machines from a set of M machine types, where the load is defined by external ERM systems,
- computer network systems with static network layout and dynamic routing strategies and random loads,
- fixed railway networks with changing time tables,
- nearly all serving processes with fixed stations and changing customer requirements.

In traditional simulation analysis such systems are modeled as "black box" models and the load is defined by parameters or in various data files. Even GPSS was used 30 years ago for defining such models. The power of this approach is determined by the quality and flexibility of the implemented interfaces for data exchange.

5 REQUIREMENTS FOR SUCCESSFUL SIMULATION IN THE WEB

5.1 Repositories of Plug & Play Models

Concerning the actual deficits of tools for development of web based user interfaces, the efforts for model design and test should be minimized at the current time. The idea of Plug & Play from computer hardware architectures will work also in web based simulation. One possible solution is a three level model repository and handling system:

- The first level provides very common, fixed models. Only external data files will change the behaviour of the models. The models are defined in the language of the used simulation system.
- The second level provides a library of predefined components. The client can define specific parameters of the components. This method is similar to well known component based systems like Arena or TAYLOR. Only the user interface and the number of forms and parameters are simplified.
- The third level allows a free definition of source code for the used simulation systems. The service of the ASP-system is limited to the execution and result analysis of the simulation.

5.2 Automatic Data Exchange

In current web based simulation environments data exchange is often reduced to manual operations, like copying text into the source code of the model or extracting results from long trace lists. Compared to professional methods of data handling in data bases or data warehouses, this level of data exchange is not acceptable for professional customers. A efficient usage of web based simulation system requires a full integration in the common data flow of the enterprise. This integration can be made by time scheduled export and import routines in ERP-systems and the simulation environment. The FTP protocol can be used for the physical transport of the data files (see Figure 2). Other protocols like HLA or CORBA are also possible, but require more development efforts.

5.3 Result Analysis with Database Functions

If the client is provided with a ERP, Data Mining or decision support system, result analysis of simulation runs is possible by importing the simulation trace files and using the integrated functions of these systems. Clients without powerful analysis tools depend on the functions provided by the web based simulation system. As demonstrated by the VisualSLX system (Wiedemann 2000) (Wiedemann 1998), this task can be solved by using databases for storing the results and calculating all aggregated values. The actual power of client-server databases also supports multi-model and multi-run comparisons. Visualization of graphical diagrams is supported by small Java applets.

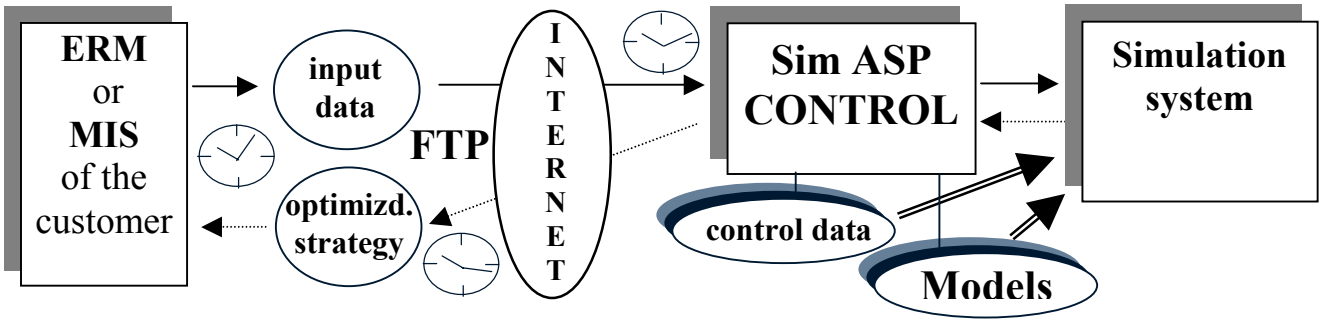

Figure 2: The Data Flow in the Simulation ASP System

5.4 Fast and Permanent Access to Simulation Control Functions

The first web based systems were often realized with CGI programs. The main disadvantages of this approach are low performance and a non-permanent connection to the simulation system. In result of the used batch mode, it was very difficult to control or interrupt a running simulation from outside. Information about the progress of the simulation was also hard to catch by CGI interfaces.

A web based simulation environment should provide a permanent connection between the client and the simulation kernel. State information and control functions must be available during the whole time of a simulation run.

6 A SIMULATION ASP-SYSTEM

Concerning the discussed requirements, an Application Service Providing (ASP) System for simulation was developed. Basic elements of this system are:

- a database for all model and simulation data,
- an object oriented modeling philosophy based on model entities and attributes,
- an universal code generator for converting the model description into a simulation program.

Details of the modeling philosophy and the code generator are presented in (Wiedemann 2000). The most important feature is the interface of the simulation environment to the web. In result of existing powerful software components for internet applications this interface is realized as combination of the VisualSLX database system and a web-server-component (see Figure 3). Here we have no CGI-interface or similar technology. Database related requests from the web are received by the Winsock-component in the VisualSLX/WEB-application and are answered immediately. Advantages of this web-server integration are:

- a very high performance in result of direct data-exchange and always open database tables,
- a long-time connection between the client and the server with continuous data flow during simulation or result processing .

Actually the SLX simulation system is used as a simulation kernel. For all code templates are stored in the database, the same code generator is used for HTML files and simulation programs.

The system can work in three modes:

- as a traditional, local stand-alone system,
- as a multi-user database in a local network,
- and as a real client server system in Intranet or Internet environments.

The first two modes are realized by traditional database forms. Application specific forms can be developed in some minutes by using latest technologies of assistant supported database design (e.g., in Microsoft Access).

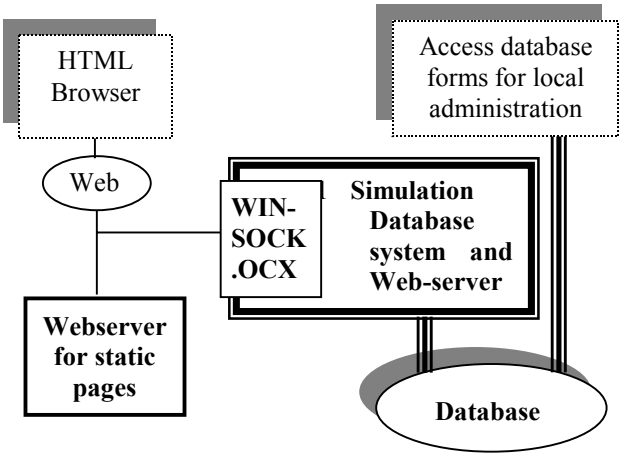

Figure 3: The Architecture of the System

The system supports all three levels of Plug & Play models according to the requirements of chapter 5.1 . The supported simulation systems are SLX and SIMPLEX III. Other systems can be integrated without large efforts.

7 WEB SPECIFIC PROBLEMS

In result of specific characteristics of the actual Internet technologies we meet some typical problems.

7.1 Web Performance and Bandwidth

Outside of the intranet the bandwidth is very often low and rapidly changing. In order to solve this problem we see a solution in parallel editing of more than one entry. For example all objects and their parameters could be offered in form fields at the same time. Checking operations are done with only one connection to the web server and all problems are reported at the same time to the user. A second option is the usage of more than one browser window and a interleave interaction mode of the user.

The best solution would be a Java-based user-interface which performs all major operations at the client side. For time reasons this solution is planned for a later version.

7.2 Multi-User Lock Problems

In the multi-user local database mode a database record is locked, when a user edits the content. During this time other users are able to see the record but they can not edit them. The Web is a system without defined sessions. Thus a user can switch off the computer or close the browser during an edit operation and the database receive no infor-

mation about the loss of the connection. Similar to modern client server system this problem can be solved by a time-out of the lock mode. In the current database this lock operation and the check for timeout is done by VisualSLX.

7.3 Run-Time Interaction

Due to the delay of information transfer from the server to the Web, the state of the simulation system and the visualization at the web based client user interface may differ for some seconds. Thus the feedback for user interactions like Break or Stopping the simulation will take two times the interchange time (from the client to the server and back), which could be sum up to 10 seconds. The main solution is revealed in the further technical improvement of the Web or a usage inside a Intranet with guaranteed quality of service. A simulation at the client side is not very useful due to the bad performance of Java.

7.4 License and Security Constraints

Web-based simulation also creates new requirements for software licensing and project management. Traditional software licenses of simulation packages only allow a single place usage. A web-based system must have the same license model like a network license. The payment of the simulation customers can be done per project or time.

Another very critical fact is also data security. If a company uses a web-based simulation system possibly sensible data will be stored in an external database. In order to provide a safe simulation study some secret data could be decrypted with a public key of the company. A special encrypt DLL will be included in the import routines of the simulation model. The private key for decryption is directly transferred between the customer and the decryption module. If the source code of the decryption DLL is validated by an external institution, the security level of the private input data for simulation will be very high.

A further improvement of security is possible **by a content scrambling** at the side of the customer. Sensitive data like customer names or product brands are replaced by random values. The conversion table is stored only at the customers computer. For the simulator there is no difference about working on a order from BMW or on data of F3234. After the simulation is finished, the results are decoded by using the stored conversion table at the customer side. Even if the network connection or the simulator is hacked by external intruders, there is no risk of loosing information, because all important data does not leave the customer system.

All security issues should be seen as very important decision factors for or against a ASP-system. If a potential customer only feels some possible security risk of giving his data in external hands, this could be a “killer fact” against the ASP-idea, where all technical and financial benefits will be without any relevance.

8 CONCLUSIONS

The perspectives of web based simulation will improve, if the current web technologies are used for a maximum of user friendliness instead of copying existing simulation systems to the web. Actually this approach will limit the capabilities of modeling large and complex systems. But the ease of use and the fast return of investments will turn this user driven approach into a very interesting way for improving and increasing the usage of simulation in commercial decision making.

The further development of this concept will enclose the following activities in the near future and is presented at (SimCo2001):

- online-animation with Java or VRML,
- graphical result analysis with database and data Mining tools via the web,
- parallel and hyper-computing with a distributed version of VisualSLX.

Important features of the developed system are:

- a flexible and open meta model for definition of application specific classes of simulation models;
- all model, meta model and simulation data are stored in databases;
- SQL as a set-oriented language for the management of the model and simulation experiments;
- one code generator for the simulation source code and the HTML code for the web interface,
- flexible architectures for data-interchange with other, complex information systems,
- interfaces to software packages concerning statistical result analysis, presentation, optimization, Data-Mining and knowledge reasoning.

With this system the needed knowledge for building a successful simulation with text based simulation systems is reduced significantly. By using the shell a user needs only common knowledge about modeling and simulation and no deep practice in the simulation language. Only the administrator needs a good SLX experience or he can modify the predefined examples.

The Simulation ASP is currently tested and will be available as an real Application Service - product. The first prototype of the new simulation environment also allows very interesting applications in the area of client-server-simulation and hyper-computing. So the presented concept could be an interesting perspective for the future development of modeling and simulation.

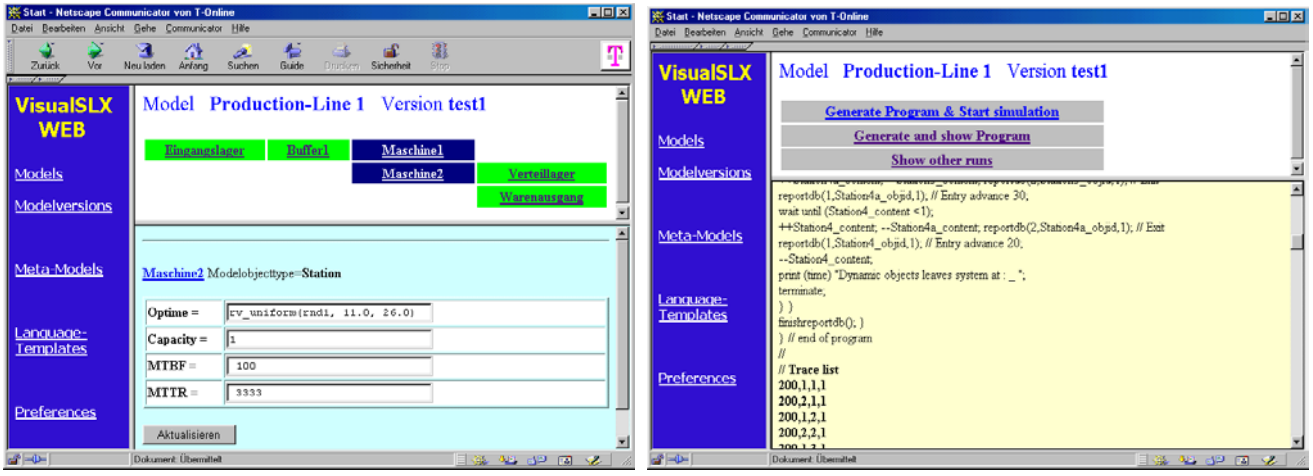

Figure 4: Examples of User Interfaces and Web Based Simulation Models

The underlying universal database structure could be seen as a first attempt for an common interchange format in the area of modeling and simulation.

REFERENCES

Fishwick, P.A. 1996. Web-Based Simulation: Some Personal Observations. *In Proceedings of the 1996 Winter Simulation Conference*, 772-779.

Fishwick, P.A., D.R.C. Hill and R. Smith (eds). 1998. Proceedings of the 1998 *International Conference on Web-Based Modeling & Simulation*. The Society for Computer Simulation International, San Diego, CA. P.A. Fishwick, D.R.C. Hill and R. Smith, 85-90.

Henriksen, J.O. 1995. An Introduction to SLX. *In Proceedings of the 1995 Winter Simulation Conference*, ed. C. Alexopoulos. 502-509. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Healy, K.J. and R.A. Kilgore. 1997. Silk: A Java based process simulation language. In *Proceedings of the 1997 Winter Simulation Conference*, 475-482.

Jasna Kuljis, R. J. Paul. 2000. A Review of web based simulation: whiter we wander?, *Proceedings of the 2000 Winter Simulation Conference*, Orlando Florida, page 1872-1881.

Wiedemann. T. 1998. Sim-Mining and SimSQL - A database oriented approach for component-based and distributed simulation, *Summer Simulation Conference 1998*, Reno Nevada.

Wiedemann. T., 2000. VisualSLX – an open user shell for high-performance modeling and simulation, *Proceedings of the 2000 Winter Simulation Conference*, Orlando Florida, 1865-1871.

SIMCO. 2001. `http://www.aedv.cs.tu-berlin.de/simco/` accessed August 28, 2001.

AUTHOR BIOGRAPHY

THOMAS WIEDEMANN is a scientific assistant at the Department of Computer Science at the Technical University of Berlin. He has finished a study at the Technical University Sofia and a Ph.D. study at the Humboldt-University of Berlin. His research interests include simulation methodology, tools and environments in distributed simulation and manufacturing processes. His teaching areas include also intranet solutions and database applications. Email : `<wiedem@cs.tu-berlin.de>`