

University of Nebraska - Lincoln

## DigitalCommons@University of Nebraska - Lincoln

---

Industrial and Management Systems  
Engineering Faculty Publications

Industrial and Management Systems  
Engineering

---

12-2000

### A web-based interface for storing and executing simulation model

Ashu Guru

*University of Nebraska at Lincoln*, ashuguru@gmail.com

Paul Savory

*University of Nebraska at Lincoln*, psavory2@gmail.com

Robert Williams

*University of Nebraska at Lincoln*

Follow this and additional works at: <https://digitalcommons.unl.edu/imsefacpub>



Part of the [Industrial Engineering Commons](#), [Operational Research Commons](#), and the [Other Operations Research, Systems Engineering and Industrial Engineering Commons](#)

---

Guru, Ashu; Savory, Paul; and Williams, Robert, "A web-based interface for storing and executing simulation model" (2000). *Industrial and Management Systems Engineering Faculty Publications*. 25. <https://digitalcommons.unl.edu/imsefacpub/25>

This Article is brought to you for free and open access by the Industrial and Management Systems Engineering at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Industrial and Management Systems Engineering Faculty Publications by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

## **A WEB-BASED INTERFACE FOR STORING AND EXECUTING SIMULATION MODELS**

Ashu Guru  
Paul Savory  
Robert Williams

Department of Industrial and Management Systems Engineering  
175 Nebraska Hall  
University of Nebraska  
Lincoln, NE 68588-0518, U.S.A.

### **ABSTRACT**

The dominance of the Internet in the development of information and communication technology has made Web-based distributed solutions increasingly attractive. Apart from providing other services, the World Wide Web is being looked upon as an environment for hosting modeling and simulation applications. SIMAN is a simulation language that allows users to simulate discrete and continuous systems. In this research, a web-based interface or toolkit has been developed for storing and executing SIMAN simulation models over the Internet. This toolkit consists of a World Wide Web interface to SIMAN and a web-accessible database for storing user models. It provides users an easy-to-use environment for developing text-based simulation models using the SIMAN simulation language. It also allows users to test the functionality of a SIMAN model using the SIMAN debugger/run controller.

### **1 INTRODUCTION**

Simulation is an indispensable tool that has proved its capability to help implement better decisions in real-world situations. It provides users the ability to accurately estimate the statistics and efficiency of various solutions and options without physically having to implement them. Though, in order to use simulation, one needs to have access to powerful computer hardware and software.

Currently, the Internet dominates the development of information and communication technology. Web-based or distributed solutions and systems are becoming increasingly attractive. The major aspects of the Internet's influence are platform independence, maintenance minimization, reusability, and interoperability. With the exponential growth of Internet communication and advancements in client/server technology, it is now

possible to share expensive computational resources over the limitations of geographical distance. The World Wide Web is being looked upon as an environment for providing modeling and simulation applications. Improvements in related technologies such as CGI (Common Gateway Interface) and HTML (Hypertext Markup Language) have allowed for the research in Web-based simulation.

In this research, a toolkit consisting of a World Wide Web interface to SIMAN is developed (Guru 2000). SIMAN is a simulation language that allows users to simulate discrete and continuous systems. The developed interface provides users an easy-to-use environment for developing text-based simulation models. The components of the toolkit are allocated to different servers. The implementation is essentially a "fat" server and "thin" client, where most of the processing is performed at the server. The clients have access to the server components using a web browser interface. This implementation is done independent of the SIMAN source code. By using this approach, the interface acts as a separate shell that interacts with the SIMAN simulation engine over the World Wide Web. In order to use the system, a user simply requires access to the Internet and an account on the web server. Apart from the essential features such as compiling, linking and executing simulation models over the World Wide Web, additional toolkit features include:

- Capability to test the functionality of a model through the use of the SIMAN debugger/run controller.
- Capability to store the simulation model in a database so that they can be easily accessed, retrieved, and updated using a user-friendly interface.

All the implementation of the developed software is done in a modular way so that the developed code can be reused for integration with other similar applications.

## 2 TOOLKIT OVERVIEW

The complete system involves a combination of three hardware and software platforms – an application server, a database server and a SIMAN server. All these servers are behind a firewall for security purposes. Figure 1 shows the layout of the three platforms and the interactions between them.

The first server is the *Application server*. An Internet user accessing the web address for the site is directed to this server. The software residing here performs the functions of authenticating the user and providing access to the user’s models in the database. It gives the user the capability to edit, add, or delete models in his or her account. When the user wants to compile, execute or debug a model, this server creates a dynamically-generated HTML document that contains the model frame, experiment frame and model ID as form elements. The document can then be “Posted” to the SIMAN server with the instructions of the desired operation.

The second server is the *Oracle server*. The Application server interacts with the Oracle server in order to provide accessibility to the user’s models. The models and the user data are stored in a database on this server using a simple two-entity schema. The first entity is “user” and has attributes: user ID, user name, email, login and password. The second entity is “model” with attributes: model ID, experiment frame and model frame. The two entities have a relationship of one-to-many (one user can have many models).

All the requests for compiling, linking, executing and debugging the models are posted to the SIMAN server.

The SIMAN simulation software that reside on this server performs the requested operation and returns the results to the user in the form of an HTML document. This is done by piping the output of the appropriate SIMAN process(es) that are invoked with the CGI script as a dynamically-generated HTML document. The arguments to these SIMAN processes are ASCII text files containing the SIMAN model and experiment frame. These files are temporarily created by CGI scripts with the data “Posted” to them through the HTML forms.

To summarize, the developed toolkit allows users to access their simulation models stored in an Oracle database over the Internet. It gives them the capability to edit, update or delete these models through HTML forms. Once saved to the database, the models can be compiled, linked, executed and debugged on the SIMAN server through the World Wide Web. The results of the three processes are then sent back to the user in the form of a standard HTML document.

## 3 SYSTEM ARCHITECTURE

A multi-stage approach was taken to analyze and design the toolkit that will act as an interface between the World Wide Web and the SIMAN simulation engine. The development process involved determining the information requirements of the intended users. System needs were analyzed and structured decisions were made. These decisions included the conditions, alternatives, actions, and rules under which a model would be developed.

Testing a simulation model’s validity and reliability to estimate the original system is an important part of the simulation process. The SIMAN software provides a very effective tool, the SIMAN debugger or run-time controller, to perform this operation. It was decided to make this run time controller accessible through the Internet.

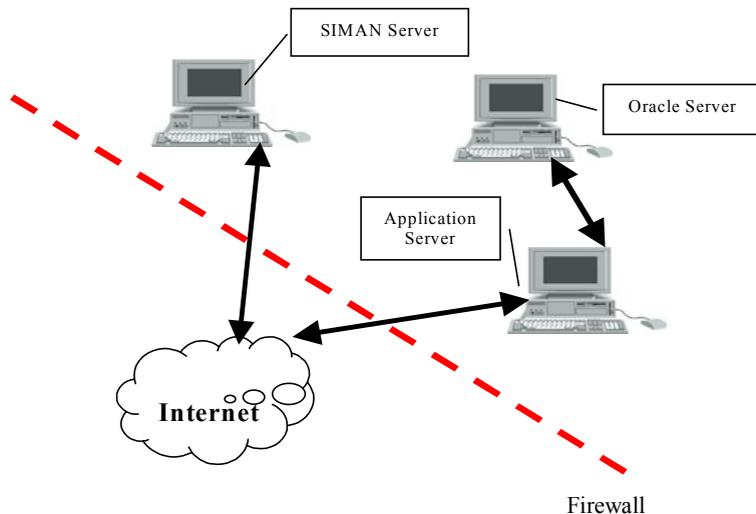


Figure 1: Hardware and Software Layout

One of the key features of the toolkit is to give a user the capability to store their models on a remote server and provide accessibility to these models from anywhere through the Internet. This facility does not require a user to use FTP or TELNET protocols. To increase the system's functionality, security, and to make the system administration's tasks easier and faster, the simulation models are stored in an online database. The simulation models can be retrieved, updated or edited by the user. As users grow, the database will form a repository and a knowledge bank of simulation models.

Based on the evaluated requirements and the decisions made, the complete system was emulated with process and data flow diagram models (Guru 2000). These preliminary models were developed to help make the development process modular and manageable. Each detail and function to be provided in the system was depicted and explained. These models formed the skeleton for the system development and implementation cycle. Even though these models were edited during the complete development cycle, they provided an indispensable aid throughout the development process. Certain conventions were followed while developing these models and data flow diagrams. These conventions are based on the recommendations made by Kendall et al. (1999). The diagramming symbols used for the data flow diagrams are based on work by Gane and Sarson (1979). Figure 2 depicts the four basic symbols and their representations in the data flow diagrams.

A simplified stepped data flow diagram (Figure 3) shows the global view of the system, its interaction with the

users (external entities to the system), as well as inter-process communications. In this diagram, the processes have been grouped together to simplify the model. To access an account and its associated simulation models, a user must present the proper username and password. Once the user has been authenticated by the system (Figure 3 - Process 1), the user is presented with a screen that contains hyperlinks to models previously stored by the user (Figure 3 - Process 2). The user has the capability to request an editing screen for a new or a previously stored model. Once the user has made a request, the appropriate screen is shown. The user can make the edits or write the new model and save the edits to the database (Figure 3 - Process 3).

After have been saved to the database, the system presents the execution screen to the user. According to the user request, the system either presents the results of compilation, linking or execution (Figure 3 - Process 4) or brings up the debugger screen for the user (Figure 3 - Process 5). From the debugger interface the user can interactively debug the simulation program. This is accomplished by stopping the SIMAN debugger process and piping the output to the user through a dynamically-generated HTML document. The state of the debugger is maintained by storing the previous debugger commands of the user as an HTML form element sent back as a part of debugger output. The debugger process is then executed again with the previous commands appended with the new ones posted by the user.

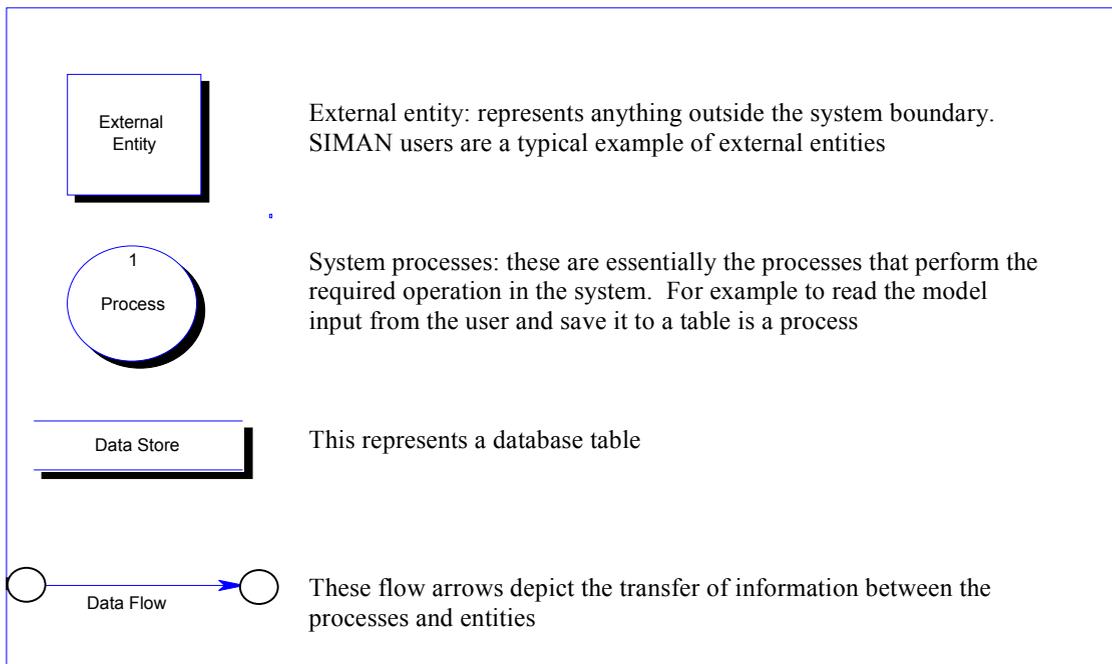


Figure 2: Basic Symbols For Data Flow Diagrams

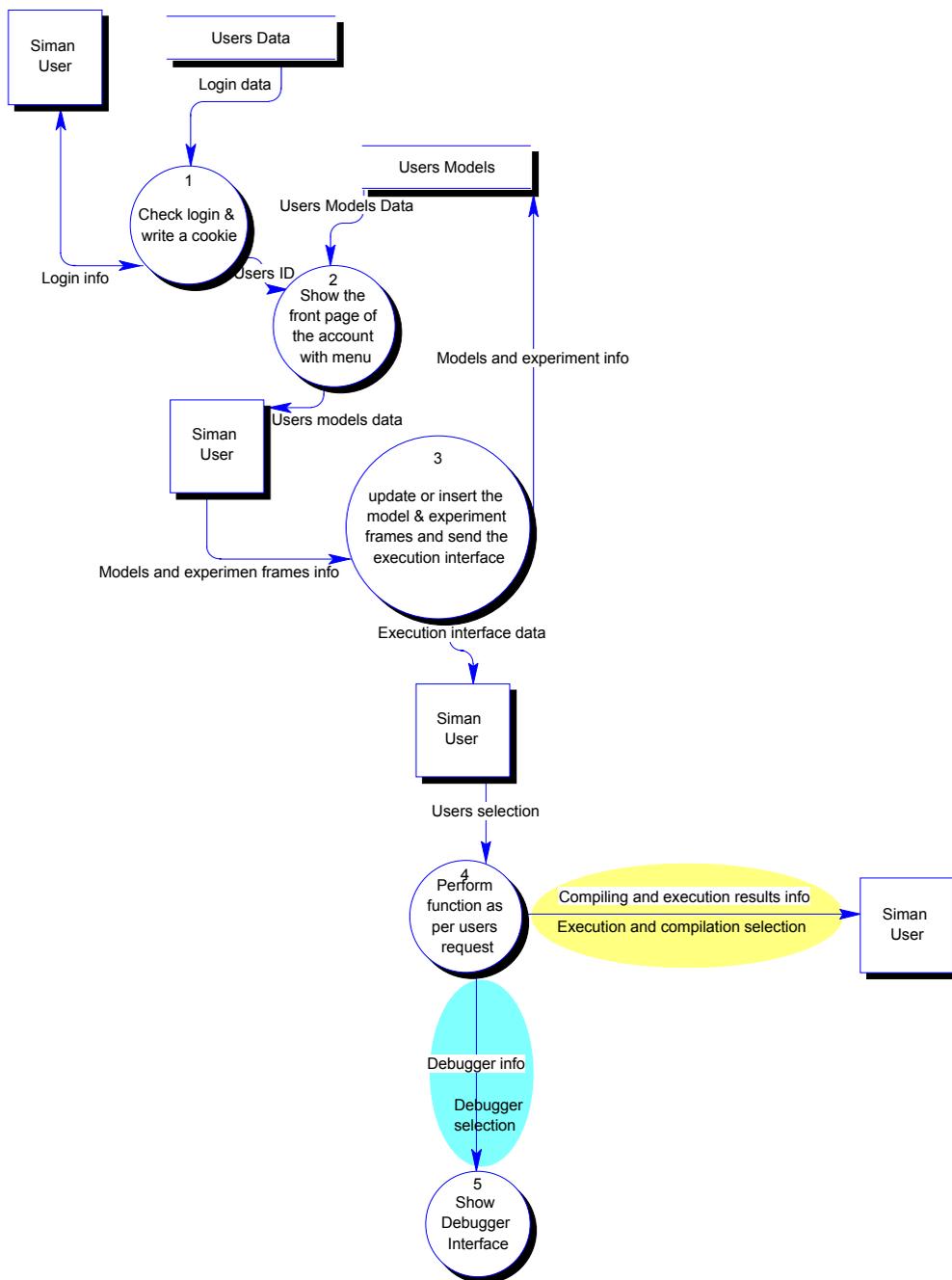


Figure 3: Overall Process Diagram

#### 4 RESEARCH CONTRIBUTIONS

The important contributions of this research are:

- (1) The developed toolkit provides the capability of executing SIMAN simulation models on a remote simulation server through the World Wide Web.
- (2) All the functionality is implemented by CGI scripting using Perl. Being written in a scripting

- language, the system is flexible enough to be ported to various operating platforms.
- (3) The developed toolkit offers advantages over classical simulation systems. For example, the modular structure of the toolkit environment allows an exchange of software tools.
- (4) The toolkit allows sharing the software and hardware resources over the Internet, independent of the user's platform. A user has access to their

account from anywhere in the world using the Internet.

- (5) The system administration is minimized through a user-friendly interface.
- (6) The toolkit provides improved security since it does not allow users to telnet into a remote server. This reduces major risks, administration efforts, and security problems.
- (7) Due to the integration of the database system to store user simulation models, the toolkit has the capability to form a repository of SIMAN models that can be searched.
- (8) The implementation of the complete toolkit has been done in a modular manner. The developed database editor is very flexible. It can be utilized to provide service to any database system that should be connected to the World Wide Web with a minimum of alterations in its configuration files.

## 5 FUTURE RESEARCH WORK

To improve the functionality of the toolkit, the following additional research is needed:

- (1) The toolkit is essentially a fat server and a thin client configuration. Since Internet technology is improving to allow executables to run on client browsers using Java, the system could be modified to allow graphical model construction.
- (2) Other services, such as the input analyzer and the output analyzer of the Arena suite, can be made accessible through the Internet.
- (3) The current system is incapable of handling SIMAN simulation models that access external input and output files.
- (4) Simulation results are currently not saved. The system can be changed to allow a user to save results into the database.
- (5) The effect of using such an environment for learning must be tested. Whether accessibility, user friendly interface and easier availability of such resources make any difference on the learning capability and interests of the users should be investigated.

## REFERENCES

- Kendall, K., and Kendall, J. 1999. *Systems analysis and design*, Fourth Edition, Englewood Cliffs, NJ: Prentice Hall.
- Guru, Ashu, 2000. A modular and integrated simulation management toolkit for storing an executing simulation models over the world wide web. M.S. Thesis, University of Nebraska-Lincoln.

Gane, C., and Sarson T. 1979, *Structured systems analysis: tools and techniques*, Englewood Cliffs, NJ: Prentice-Hall.

## AUTHOR BIOGRAPHIES

**ASHU GURU** is a Doctoral Student with the Department of Industrial and Management Systems Engineering at the University of Nebraska-Lincoln. He earned his Bachelor's degree in Mechanical Engineering from Delhi College of Engineering, Delhi, India and his M.S. in Manufacturing Systems from the Industrial Engineering department at the University of Nebraska. His research interests include database management systems and discrete event simulation. His email address is <ashuguru@hotmail.com>.

**PAUL SAVORY** is an Associate Professor in the Industrial and Management Systems Department at University of Nebraska. Having degrees in industrial engineering, operations research, and computer science, he has taught numerous courses and consulted in the areas of computer simulation, operations research, and statistics. His email address is <psavory@unl.edu>.

**ROBERT WILLIAMS** is an Associate Professor with the Department of Industrial and Management Systems Engineering and the Center for Nontraditional Manufacturing Research at the University of Nebraska-Lincoln. Dr. Williams' areas of research and teaching interest include traditional and nontraditional manufacturing processes, computer-aided manufacturing, and rapid prototype development. His email address is <ierpwil1@engunx.unl.edu>.