

Interfaces to Enhance User-Directed Experimentation with Simulation Models of Discrete-Event Systems

Sara T. Hewitt
TRW, Rosslyn, Virginia

Jeffrey W. Herrmann
Department of Mechanical Engineering
University of Maryland, College Park, MD 20742, USA
jwh2@eng.umd.edu

Keywords: Discrete event simulation, user directed exploration, discrete event systems.

Abstract

This paper describes a learning historian to improve user-directed experimentation with simulation models of discrete event systems. In user-directed experimentation, an analyst conducts simulation runs to estimate system performance. Then the analyst modifies the simulation model to evaluate other possibilities. An important characteristic is the ad hoc nature of the experimentation, as the analyst forms and runs new trials based on the results from previous trials. Through user-directed experimentation designers compare alternatives and students learn the relationships between input parameters and performance measures. Recording and reviewing previous trials while using simulation models enhances their benefits, transforming trial-and-error into learning. The learning historian combines a graphical user interface, a discrete event simulation model, and dynamic data visualization. Usability studies indicate that the learning historian is a useful tool because it allows users to concentrate more on understanding system behavior than on operating simulation software.

INTRODUCTION

Modeling is an important component of engineering design used for training purposes, to save time and money, to optimize a system, to predict performance, to enhance understanding of system behavior, and to examine worst-case scenarios. Discrete event simulation is often used when the system is too complex to evaluate analytically. Typical applications include queueing systems, manufacturing systems, and transportation systems. Animation is useful for understanding the behavior of the objects within the system. Model exploration allows users to gain insight into system performance across various scenarios and to compare alternatives. In user-directed experimentation, an analyst conducts simulation runs to estimate system performance and then modifies the

simulation model to evaluate other possibilities. An important characteristic is the ad hoc nature of the experimentation, as the analyst forms and runs new trials based on the results from previous trials.

User-directed experimentation with modern simulation software is a difficult task, however, because the software has limited capabilities to support experimentation. The user interfaces for simulation software are excellent tools for building sophisticated models, viewing animation, and performing other functions not related to user-directed experimentation. Although capabilities vary between applications, generally, the analyst must tediously modify values in the simulation model, view the large results file, and find and record the output values of interest. Better techniques are needed.

This paper describes an application (the Learning Historian) that works with simulation software to facilitate learning about system behavior by making it easier to run trials and visualize the results. This paper proceeds as follows: Section 2 describes related work on other simulation-based learning environments. Section 3 describes the Learning Historian. Section 4 describes a study with students using the tool. Section 5 concludes the paper.

SIMULATION-BASED LEARNING ENVIRONMENTS

In previous work, researchers at the Institute for Systems Research developed simulation-based learning environments for a variety of applications.

Simple

The application called Simulated Processes in a Learning Environment (Simple) [Rose *et al.*, 1998] uses VisSim, a commercial simulation package designed for simulating dynamic, continuous variables, where the output variables change as a function of time and the values of input parameters can be changed during the simulation. In an example of simulating vacuum pump technology used in

semiconductor manufacturing (shown in Figure 1), the user can start pumps and open valves while the system runs, thereby changing the state of the system. As a result of the chemical reactions within the system and the states of the valves and pumps, the pressure in the various chambers will change according to mathematical functions of the underlying processes. The graph at the top of Figure 1 shows the pressure as a function of time. The lines at the bottom of the figure show when the user opened valves and started pumps. Studies showed that students found the history mechanism valuable when they used the simulation and shared their results with others [Plaisant *et al.*, 1999].

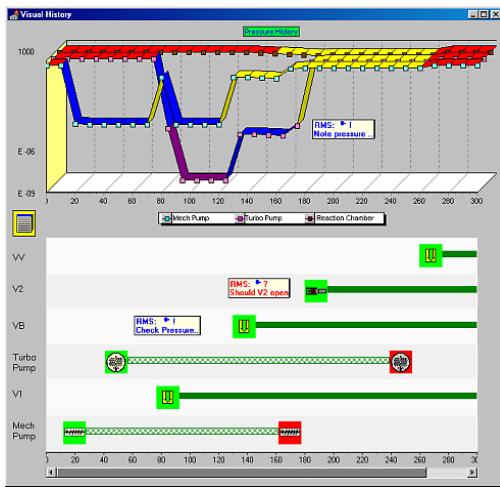


Figure 1. SimPLE

Learning Historian for Manufacturing Systems

The Learning Historian for Manufacturing Systems (LHMS) is a Java application that allows user-directed experimentation with a discrete-event simulation model of a production line [Chipman *et al.*, 2001]. As shown in Figure 2, the user sets input variables using slider bars in the control frame and runs the simulation model. The results of each subsequent run appear in the timeline frame and are added to a results data set. The user can work with an interactive scatter plot visualization tool called StarDOM (Figure 3). StarDOM allows users to review the results data set by easily constructing two-dimensional displays that can be filtered interactively using controls that match the fields of the dataset.

THE VISUAL BASIC LEARNING HISTORIAN

The current Learning Historian is a Visual Basic application that has three major components: the user interface, Arena, and a dynamic display tool available from Spotfire. The Learning Historian exploits the Visual Basic interface available in version 4.0 of Arena, a commercial

discrete event simulation software sold by Rockwell Automation. The Learning Historian works with any Arena model that contains a Visual Basic for Applications (VBA) module that can read the experiment file and modify the variable values appropriately.

The Learning Historian has four frames: the home frame (Figure 4), the input/output frame (Figure 5), the experiments frame (Figure 6), and Spotfire (Figure 7). A student using the Learning Historian interacts directly with only the three Visual Basic frames and Spotfire. To begin using the Learning Historian, the user selects an Arena model, which the Learning Historian then opens and runs to identify the names of the input and output variables. The user can open the results saved from a previous session as well.

The user selects the input and output variables of interest using the input/output frame and then defines a set of one or more trials in the experiments frame. For each trial the user specifies values for each selected input variable. When the user clicks on "Run Trials," the Learning Historian iterates over the set of trials. The application writes the variable names and the user-defined values to the experiment file. Arena runs the model, which reads this file, modifies the input variables, simulates the system, and produces an output file. The Learning Historian then reads the output file to find values for the average (and half-width of the confidence interval) of the user-selected output variables. These values are added to the results file. These results are shown as a table on the home frame (Figure 4). The user can define and run more trials, run Spotfire, or save the results of the experiments for a future session.

When the user clicks on "Run Spotfire," the Learning Historian opens Spotfire, which reads the results file. The user can then employ Spotfire to filter the results and display a variety of charts and graphs, as shown in Figure 7. Note that these results include the confidence intervals associated with output performance measures.

CLASSROOM EXPERIENCE

Students recruited from a production management class participated in an exercise that used the Learning Historian. (These students were senior undergraduate students majoring in mechanical engineering.) Participation was not mandatory, and it was made clear that the exercise was not graded and that their feedback and suggestions would help improve the Learning Historian. Students read a one-page description of the manufacturing system being simulated and definitions of key variables. The system was a fast food counter in the engineering building. The students (working in pairs) were provided with step-by-step instructions for how to complete a series of questions. The first tasks were simple tasks requiring the users to adjust input parameter values and execute trials. These helped the users become familiar with the interface. The remaining tasks were more

complex as they required the student to conduct additional trials and draw a simplified graph showing the relationship between batch size and cycle time. One weakness of user-directed experimentation is that some students attempted to reach a conclusion about relationships with too few trials.

Students found the user interface easy to use, but suggested some improvements for the Spotfire program and for the Learning Historian, including adding a tutorial or a wizard and streamlining the instructions. Those with simulation experience (a minority among the participants) said that the Learning Historian was simpler than other programs and it was easier to make changes.

SUMMARY

The Learning Historian is a user interface that enhances user-directed experimentation with a simulation model by making it faster and easier for users to run trials, automatically recording the results, and incorporating a tool for visualizing the results. Thus, the Learning Historian is a useful tool for helping students learn, through active experimentation, about how important parameters of a complex system affect system performance. The interactive nature of the tool helps both students and expert analysts to experiment more easily than using the software directly, while the visualization software helps them see patterns and relationships. The application shields the user from the complexity of the simulation software and the sophistication of the simulation model, which helps the user focus on learning (and not programming). The user does not have to know anything about the simulation software to use the Learning Historian. Although this implementation works with Arena simulation models, extending the Learning Historian to work with other simulation software is feasible.

ACKNOWLEDGMENTS

The National Science Foundation supported this work through the Collaboration for Industrial and Systems Engineering. The participants are North Carolina A&T State University, University of Maryland, and Georgia Institute of Technology. Northrop Grumman provided additional support through the Institute for Systems Research Industrial Fellowship Program. The research was conducted in the facilities of the Computer Integrated Manufacturing Laboratory, a constituent lab of the Institute for Systems Research. The authors thank Anne Rose for her help designing the interface.

REFERENCES

- Chipman, G., C. Plaisant, S. Gahagan, J. Herrmann, S. Hewitt, and L. Reaves, 2001. "Understanding Manufacturing Systems with a Learning Historian for User-Directed Experimentation." CS-TR-4243, Department of Computer Science, University of Maryland, College Park.
- Plaisant, C., A. Rose, G. Rubloff, R. Salter, and B. Schneiderman, 1999. "The Design of History Mechanisms and their Use in Collaborative Educational Simulations." CS-TR-4027, Department of Computer Science, University of Maryland, College Park.
- Rose, A., D. Eckard, and G.W. Rubloff, 1998. "An Application Framework for Creating Simulation-Based Learning Environments." CS-TR-3907, Department of Computer Science, University of Maryland, College Park.

BIOGRAPHIES

Sara T. Hewitt earned a M.S. in mechanical engineering from the University of Maryland, College Park. She is now working as a Systems Engineer for TRW in Rosslyn, Virginia. Her research interests include discrete-event and continuous simulations, methods of model verification and applications to facilitate the simulation process.

Jeffrey W. Herrmann is an associate professor at the University of Maryland, where he holds a joint appointment with the Department of Mechanical Engineering and the Institute for Systems Research. He is the director of the Computer Integrated Manufacturing Laboratory. He is a member of INFORMS, ASME, IIE, and SME. He earned a B.S. in applied mathematics from Georgia Institute of Technology. He received his Ph.D. in industrial and systems engineering from the University of Florida. He has led research projects supported by NSF, NIST, ONR, the Semiconductor Research Corporation, the U.S. Air Force, the U.S. Army Tank-Automotive Command, Lockheed Martin Missiles and Space, Harris Semiconductor, Northrop Grumman Electronic Sensors and Systems Division, Black & Decker, and other manufacturers in the state of Maryland. His current research interests include the design and control of manufacturing systems, the integration of product design and manufacturing system design, and decision-making systems in product development.

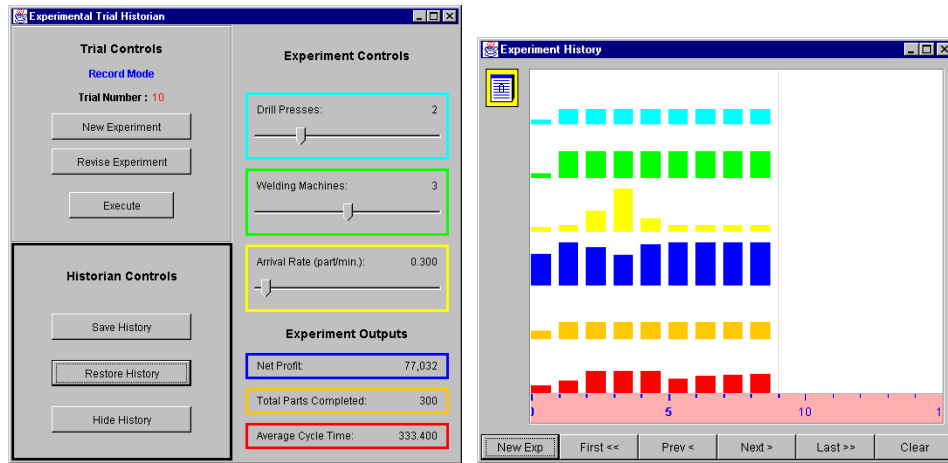


Figure 2. The LHMS control frame (left) and timeline frame (right).

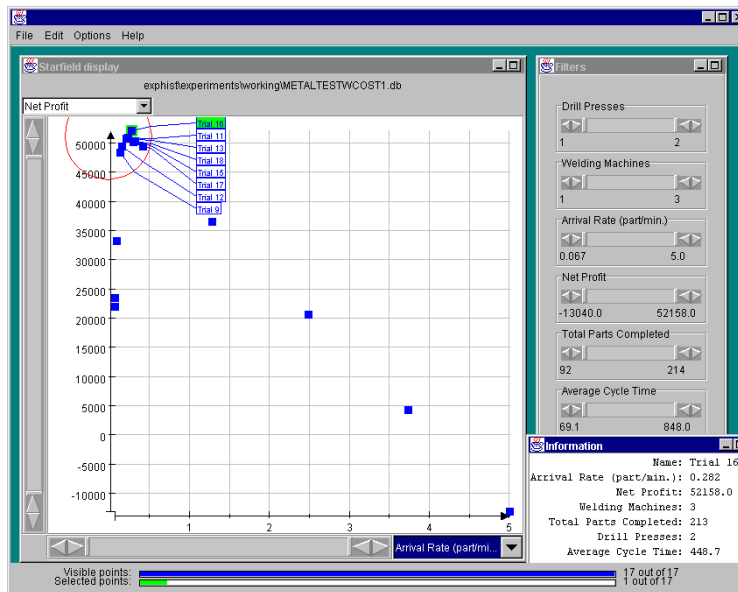


Figure 3. StarDOM

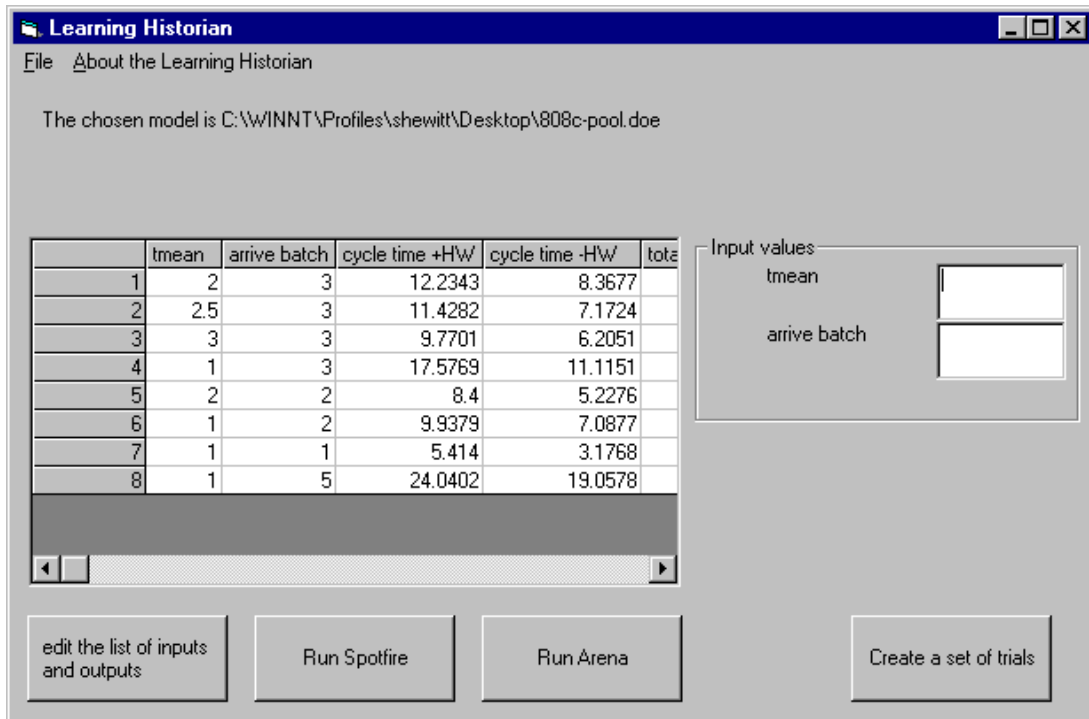


Figure 4. The home frame

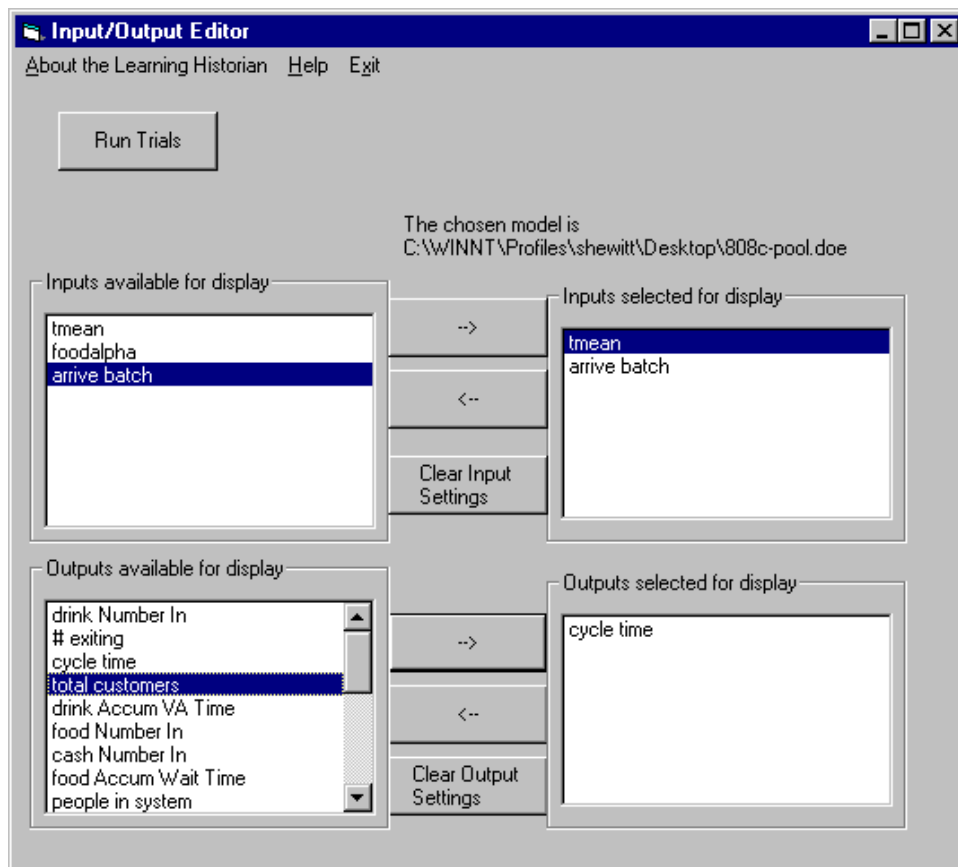


Figure 5. The input/output frame

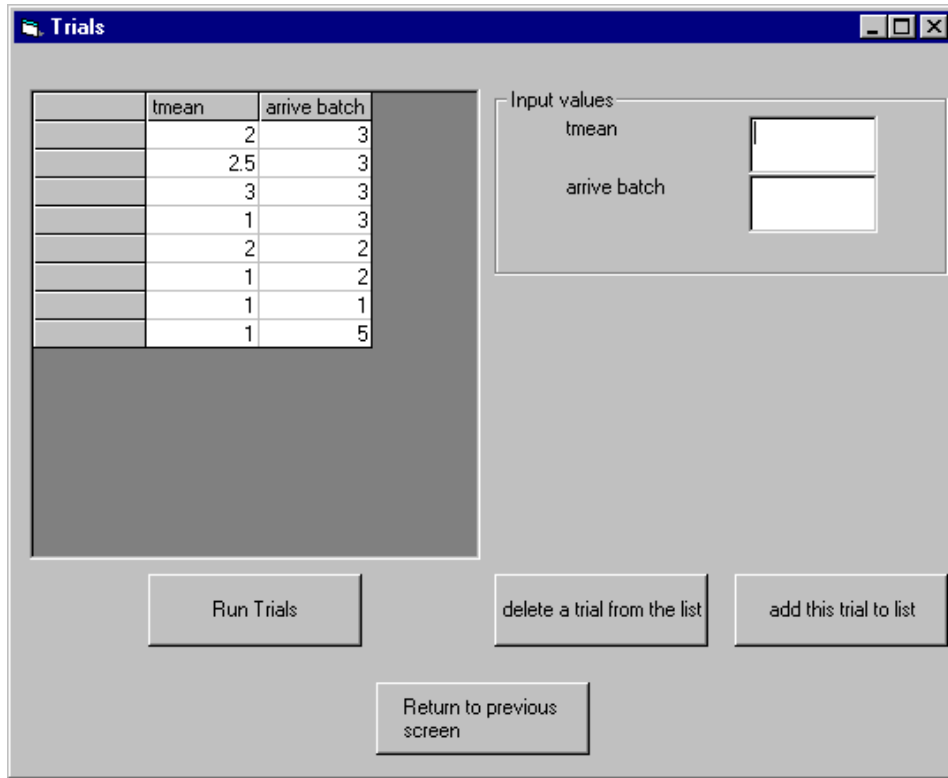


Figure 6. The experiments frame

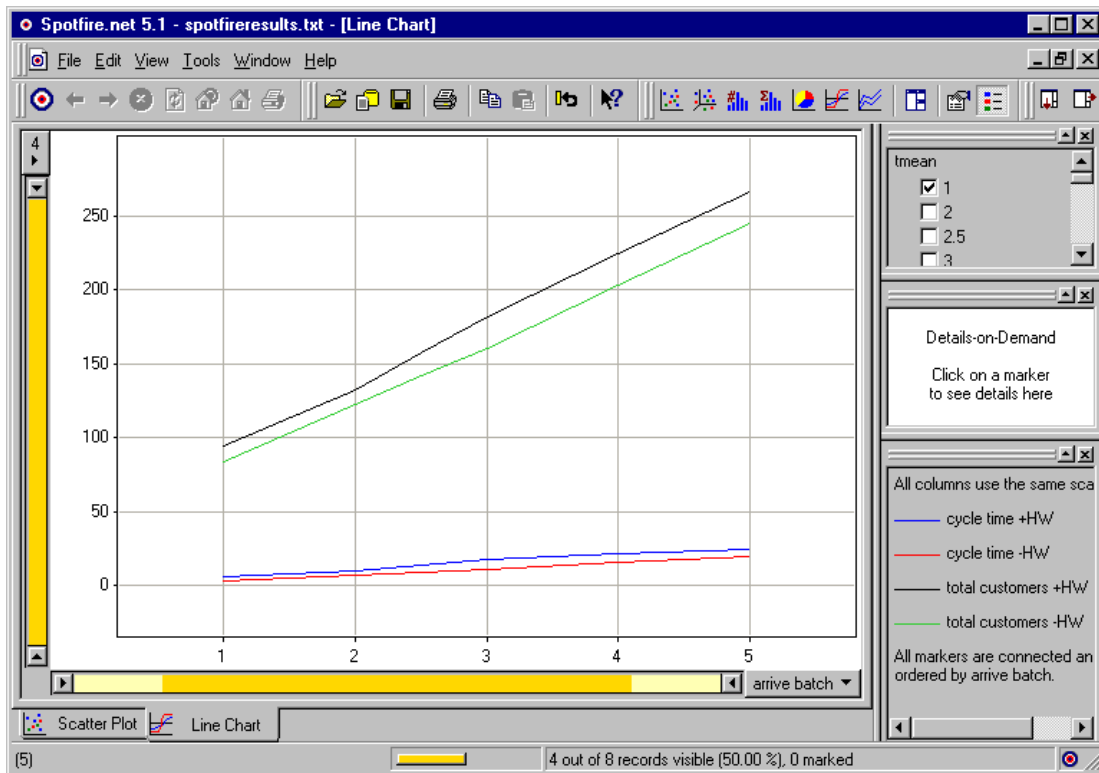


Figure 7. Spotfire