*Proceedings of the 1997 Winter Simulation Conference*
*ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson*

# A USER INTERFACE TO SUPPORT EXPERIMENTAL DESIGN AND DATA EXPLORATION OF COMPLEX, DETERMINISTIC SIMULATIONS

L. Tandy Herren
Pamela K. Fink
Christopher J. Moehle

Medical Science Systems, Inc.
100 N.E. Loop 410, Suite #1350
San Antonio, Texas 78216, U.S.A.

## ABSTRACT

Simulations are designed to emulate a system or process within a certain set of specified assumptions. Such simulations can then be used as experimental platforms for exploring a system's or process's behavior under a variety of circumstances. Experiments are conducted by systematically varying the inputs to the simulation model, collecting the model outputs, analyzing the resulting data, and using the insights gained from the analysis to formulate new experiments and/or to answer questions concerning expected behavior of the system or process under study. As models become increasingly complex, in order to learn the most from the least number of runs, exploration of these models' behavior must be systematic and focused. Careful planning and experimental design must be done in order to efficiently and effectively use large, complex models to answer key questions. The Experimental Design and Analysis Simulation Interface supports this process for large, complex, deterministic models.

## 1 INTRODUCTION

Simulations are designed to emulate a system or process within a certain set of specified assumptions. They are usually developed for a specific purpose, such as to answer questions about how the system or process will behave under certain conditions and how altering those conditions will alter the behavior. As a result, simulations serve as a platform for experimentation, allowing an individual to specify variations in the input (e.g. define a scenario) and to observe the results as output. However, if the simulation involves the modeling of a system or process that has more than a few inputs and outputs, with more than a few possible variations in those inputs and outputs, then developing the inputs, running the series of simulations, and exploring the outputs systematically can be a very difficult task. The problem is combinatoric. For example, even a simulation with only 10 inputs that can each be varied in 2 different ways results in 1024 possible input scenarios, each of which represents a different simulation run. If each simulation run outputs 10 different parameters of interest then, in order to completely explore system behavior, there are over 10,000 different data points that need to be examined for even this simple model. Thus, to effectively use a complex model, support is needed for both input generation and output exploration and analysis.

Support for model input generation is minimal in most commercial packages. If the simulation being developed is based on a stochastic model, then there are a few commercial add-on products that help to produce appropriate statistical distributions from example data (Jankauskas and McLafferty, 1996; Law and McComas, 1996). Many commercial modeling packages do not even provide file import facilities (Kuljis, 1996), much less a mechanism for reading a series of input scenarios from a database, or a tool for helping to generate an appropriate factorial experimental design.

Support for data/results analysis can be divided into two classes: 1) statistically-based charting and plotting and 2) visually-based animation. Most commercial simulation development tools provide support for collecting and storing the output data values of a run so that they can be graphed, displayed in tables, or sent to a statistical package for analysis. Through such analysis a user can, for example, identify bottlenecks in a manufacturing process and determine what effect various solutions, such as improving machine up-time or increasing the delivery rate of materials, might have on that bottleneck in terms of service time, as well as overall system performance. These tools also help to estimate the output distribution of the parameters in a stochastic model. Some commercial simulation tools provide support for run-time animation of the modeled system (e.g. Taylor II (King, 1996), ProModel (Benson, 1996), Arena (Markovitch and Profozich, 1996), and Extend (Krahl, 1996)). Such systems can display, in a 2-D and even a 3-D visual representation of the physical model, providing a run-time display of what is happening during a simulation. This allows the user to observe, as the model is running, how a system behaves under a given set of inputs. From such interfaces, queue lengths to the various servers can be observed, routings and arrival times can be watched, and a general sense of how the system is working can be obtained, but only for one run at a time. Thus, it is difficult with this kind of a data/results analysis facility to compare how variations in input can affect the output across numerous runs.

In summary, these standard analysis tools are not sufficient for managing all of the data that are involved in answering questions about a large, complex system. The ability to actually observe the system during the simulation may be very useful when a detailed analysis of a particular scenario is needed. However, it is not possible to observe thousands, or even hundreds of individual runs, and be able to discern what effects altering the inputs to the system has on key system behaviors. Storage of the outputs of the various runs and using a statistical package to analyze results can help, but this approach requires that the user define the relationships of interest in an *a priori* manner. Oftentimes there may be interesting patterns in the data, but they risk going undetected because the user does not know how to characterize their attributes sufficiently in advance. Statistical analyses need to be focused and do not, therefore, work very well in situations where the dimensionality of the data is extremely high and where aspects of the data that might contain interesting relationships are not known.

In situations where there is a large number of inputs, each with high dimensionality, the number of possible combinations of inputs and outputs to a simulation can range into the billions and even the trillions. It is not, therefore, reasonable to explore all possible combinations of inputs, and to examine all of the resulting outputs statistically, in order to learn what needs to be learned from such models. As models become increasingly complex, exploration must be more focused and well-targeted in order to assess their behaviors efficiently and effectively. This can only be achieved through careful planning and experimental design, and with support for principled database mining. Thus, in order to explore a large, complex simulation in a cost effective manner, support for the user is needed for both 1) designing the series of experiments needed to generate the data that will contain the key behaviors of interest and 2) exploring the resulting data so that the patterns and effects of inputs on outputs can be found.

The Experimental Design and Analysis Simulation Interface (EDASI) was developed to support experimental design and analysis of large, complex, deterministic simulation models. This interface addresses both the problem of setting up the appropriate input parameters to the model so that the needed simulation runs are made to answer the key questions and of exploring the resulting data once the model has been run. The following section describes an example from a biological domain to introduce one implementation of the EDASI. Then the EDASI is described in detail. Finally, we discuss the impact of the EDASI on broadening the utility of simulation models.

## 2 AN EXAMPLE APPLICATION AREA: PERIODONTAL DISEASE

Approximately 30% of the U.S. population has periodontal disease. This disease is progressive and, if not managed properly, ultimately results in tooth loss. The etiology of the disease is very complex, however it can be traced to an overgrowth of the natural flora in the mouth. The patient’s inflammatory and immune system attempts to counter the growing bacterial challenge but, in doing so, the cells generate products that break down the collagen matrix that attaches the gums to the teeth. As this attachment degrades, the underlying bone begins to break down. Finally, the gums recede to such an extent that the tooth becomes loose and is lost. Brushing, flossing, and regular dental visits can keep the bacterial growth in check and control, if not prevent, the disease process in most people. However many people, such as smokers, do not benefit from these elementary measures sufficiently to prevent the onset and progression of periodontal disease.

We developed a simulation of periodontal disease to test a variety of hypotheses about how to attenuate periodontal disease progression. An EDASI was developed to manage the experimentation and data exploration/analysis using this model. The simulation model contains twelve different cells lines, has 126 input parameters, and 55 output parameters. Most of the input parameters may assume any value between 0 and 5 in increments of 0.25. Some input parameters have a more limited range, but none has a range of less than 10 variations. The model is run for two years at 6 hour intervals. The output parameter values are collected monthly and saved in the EDASI over the course of the two year simulation period. Thus, to sample just a range of two possible input values for each of the 126 input parameters (e.g. yes/no or on/off) constitutes $2^{126}$ different input scenarios, each of which represents a unique simulation run. Furthermore, each output scenario generates approximately 1500 data points.

It should be noted that, unlike many discrete-event simulation models, this model is not stochastic. Rather, it is deterministic in that the outputs will be the same for each simulation run given the same set of inputs. Neither the inputs nor any of the system behaviors are statistically-based. Thus, the issue concerning the generation of sets of inputs is to ensure that all relevant scenarios have been produced, and not that the sets of inputs model some distribution base on some real-life dataset (e.g. arrival rates, service rates, etc. (Leemis, 1994)). Furthermore, analysis of the output centers on determining how changes in the values of the input parameters (e.g. treatment regimens, patient attributes) affect the behavior of the system (e.g. disease process) and, consequently, the output of the system, rather than trying to deduce an output distribution for the system (e.g. hourly production level, machine utilization, etc. (Kelton, 1994)).

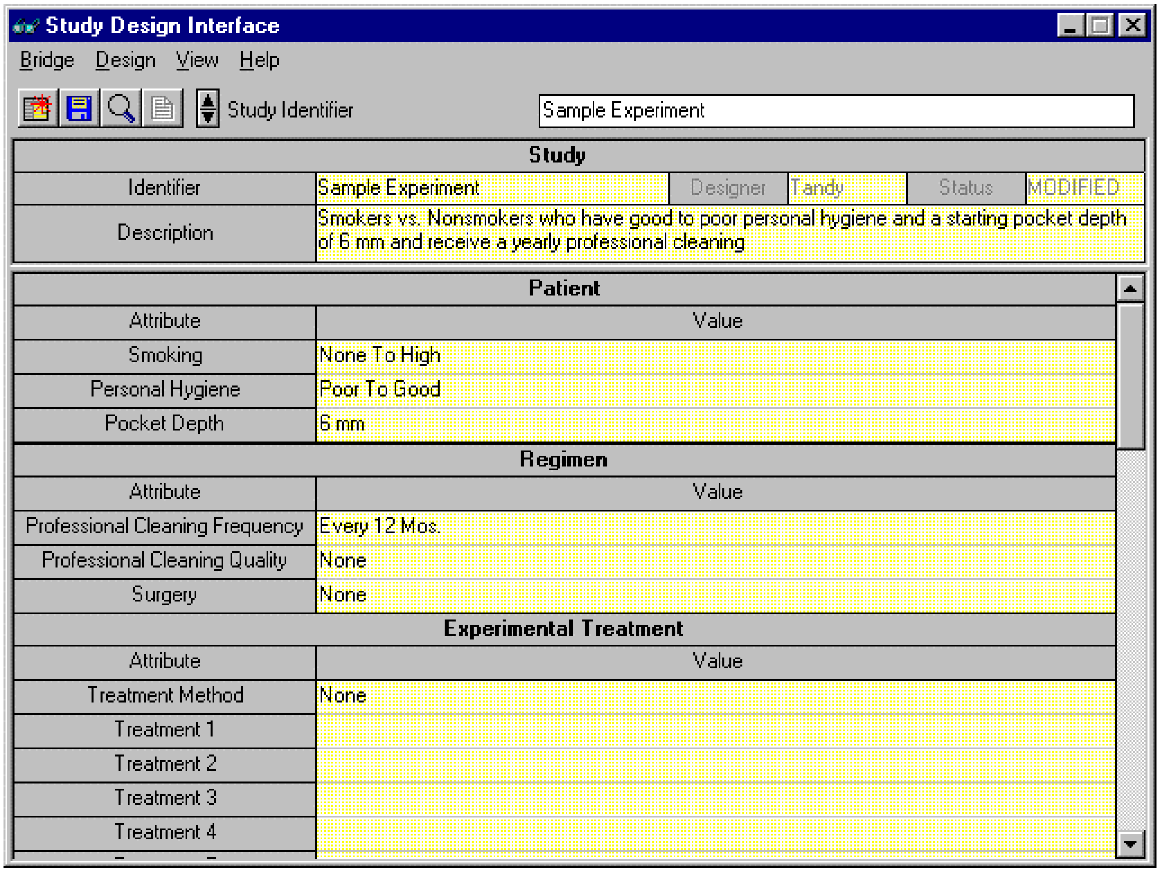
## 3 THE EXPERIMENTAL DESIGN AND ANALYSIS SIMULATION INTERFACE

The Experimental Design and Analysis Simulation Interface consists of three major modules: the Study Design Interface, the Simulation Results Database, and the Data Exploration Interface.

### 3.1 Study Design Interface

The Study Design Interface supports the functions that help the user efficiently set up and run a model, including the design of experiments and the set-up of the input data, the management of the actual simulation runs, and the storage of the results of the runs into the Simulation Results Database. Figure 1 provides an example of this interface for the periodontal disease area. First, the Study Design Interface supports the use of English-language descriptions of the model parameters, rather than numerical inputs which are often confusing. For example, the periodontal disease model supports a range between 0 and 1 to represent the level of smoking engaged in by the patient (0 is non-smoker, 1 is heavy smoker), but only three semantic categories are actually needed to capture the range in variation that is of interest: nonsmokers, moderate smokers, and heavy smokers (i.e., greater than half a pack of cigarettes per day). The Study Design Interface allows the user to select among these three labels to more naturally define the subject’s smoking habits. English-language descriptions help the user to understand the import of parameter variations more readily and to select the variations that are most likely to affect the question that the user is attempting to answer with the simulation.

Second, the Study Design Interface can help the user conceptualize and conduct experiments using the simulation. The user selects ranges of parameter values to vary in the study using the Study Design Interface. Once the user selects the range of values for each parameter of interest, the Study Design Interface generates all factorial combinations of those variables. The user then elects to run all factorial combinations or may reject some irrelevant or impossible combinations to form a partial factorial study. The Study Design Interface then systematically runs the simulation model with each combination of parameter values, retrieves the simulation output, and stores the results in the Simulation Results Database for future exploration and analysis. For example, in the periodontal disease simulation, a user may want to explore how smoking interacts with a number of other patient attributes and/or therapies. In periodontal disease, these attributes include the level of personal hygiene (i.e. brushing and flossing) of the subject and the regularity of professional cleanings that the subject receives. When the user selects these attributes and appropriate ranges, the interface lists them across their semantic values, such as brushing once vs. twice per day for personal hygiene, and visiting a dentist quarterly, every six months, or yearly for professional cleanings, and crosses them with the three smoking values to produce a factorial experiment for all possible combinations (a total of 18 possible combinations in this case).



Study Identifier: Sample Experiment

| Study | | | | | |
|---|---|---|---|---|---|
| Identifier | Sample Experiment | Designer | Tandy | Status | MODIFIED |
| Description | Smokers vs. Nonsmokers who have good to poor personal hygiene and a starting pocket depth of 6 mm and receive a yearly professional cleaning | | | | |

| Patient | |
|---|---|
| Attribute | Value |
| Smoking | None To High |
| Personal Hygiene | Poor To Good |
| Pocket Depth | 6 mm |

| Regimen | |
|---|---|
| Attribute | Value |
| Professional Cleaning Frequency | Every 12 Mos. |
| Professional Cleaning Quality | None |
| Surgery | None |

| Experimental Treatment | |
|---|---|
| Attribute | Value |
| Treatment Method | None |
| Treatment 1 | |
| Treatment 2 | |
| Treatment 3 | |
| Treatment 4 | |

Figure 1: Example of the Study Design Interface for a Model of Periodontal Disease

The Study Design Interface then stores the input combinations in the Simulation Results Database and runs the experiment by launching the simulation for each combination of parameter values. After each simulation run, it collects the output values from the simulation and stores them in the Simulation Results Database.

### 3.2 Simulation Results Database

The Simulation Results Database is designed to store the model input and output values for each run conducted by the Study Design Interface. The complexity of the data model is often extraordinarily high because the number of model inputs and outputs can range into the hundreds, thousands, or even more. Also, in a complex simulation, the user is often interested in intermediate results as well as the final outcome. So, not only are there potentially many output parameters of interest, the Simulation Results Database may need to store the values of the output parameters at various time intervals during the course of the simulation. Therefore, the Simulation Results Database is generally quite large and complex. For the periodontal disease model, the Simulation Results Database contains the input parameter values for each of the 126 input parameters, and the output values for each of the 55 output parameters at each month over the course of 24 months. Thus, one simulation run generates a record with nearly 1500 data points.

The design of the Simulation Results Database must carefully consider the user's needs. During implementation, decisions are made about which output parameters are necessary to fully capture the range of model behaviors that are likely to be of interest to the user and which points in time over the course of a run are particularly informative. Values for the model behaviors of interest are stored in the Simulation Results Database at each of the relevant points in time.

The Simulation Results Database is the heart of the EDASI. Simulation users generally interact with a simulation on a run-by-run basis, examining the results of a single run before designing the next run. The Simulation Results Database allows a user to save the relevant data from every simulation run made so that it can be further explored and analyzed at a later date and/or compared with other simulation runs. Many, sometimes thousands, of runs are stored in the Simulation Results Database, providing the user access to a set of data representing a large portion of a complex system's potential behavior under the set of conditions of interest. If a run is of particular interest, the user can run the simulation to examine the pattern of results at an even more detailed level. But, in general, the user can evaluate and discard numerous hypotheses easily and quickly without resorting to the difficult and time-consuming process of setting up and running the model for each combination of parameters of interest. This helps the user focus on promising avenues more quickly.

### 3.3 Data Exploration Interface

Because the amount of data generated by complex model simulations can be enormous as well as highly multidimensional, it is difficult for a user to make sense of the resulting data and to answer the questions that the model has been designed to help answer. The Data Exploration Interface allows the user to explore the results of simulation runs, either individually or across any specified set of criteria. Because the results of every simulation run are stored in the Simulation Results Database, they are available for examination in any combination.

The presentation format in the Data Exploration Interface depends on the form and quantity of the data generated by the model runs. The tool has been designed to allow a user to explore the data in search of patterns and correlations, thus providing a means to mine the database. For example, it can graph a variable, such as attachment loss in the case of periodontal disease, over time for different experimental groups, or it can chart the outcome for all variables, from attachment loss to the number of cells in a particular state, at a single point in time for a single experimental group. This is illustrated in Figure 2.

To illustrate, the user might ask the system to display attachment loss in increments of 0.1 mm across the 13th to the 24th month of the simulation and to indicate which losses are associated with smokers who have regular professional cleanings. The Data Exploration Interface queries the Simulation Results Database and retrieves the relevant data. It then displays counts for each category of loss, i.e., each 0.1 mm of loss, for each of the 12 months and uses different colors to highlight those results that are associated with smokers who see their dentist regularly. It also displays, upon request, descriptive statistics, such as the mean attachment loss for smokers vs. nonsmokers at each month. The user may elect to see results for any of the 55 output parameters at any point in time conditioned on any of the input parameter values. This allows the user to see if any patterns exist, such as whether or not more regular professional cleanings results in less disease progression for these types of patients.

The Data Exploration Interface can also map the results onto an anatomical representation. Using heuristics, for example, supplied by experts in periodontal disease, the attachment loss computed by the simulation can be converted into the probable loss rates across all sites in the mouth. Thus, an attachment loss of 1 mm may result in a 50% likelihood that probing sites that have an original depth of 4 mm deteriorate by 1 mm, a 25% likelihood that these sites only deteriorate by 0.5 mm, and a 25% likelihood that these sites do not actually exhibit a clinically detectable change. A graphic of the mouth is displayed to depict these heuristic relationships to help the user understand the probable clinical manifestation of the simulation results in terms of number and location of sites experiencing the various levels of attachment loss over a given period of time.

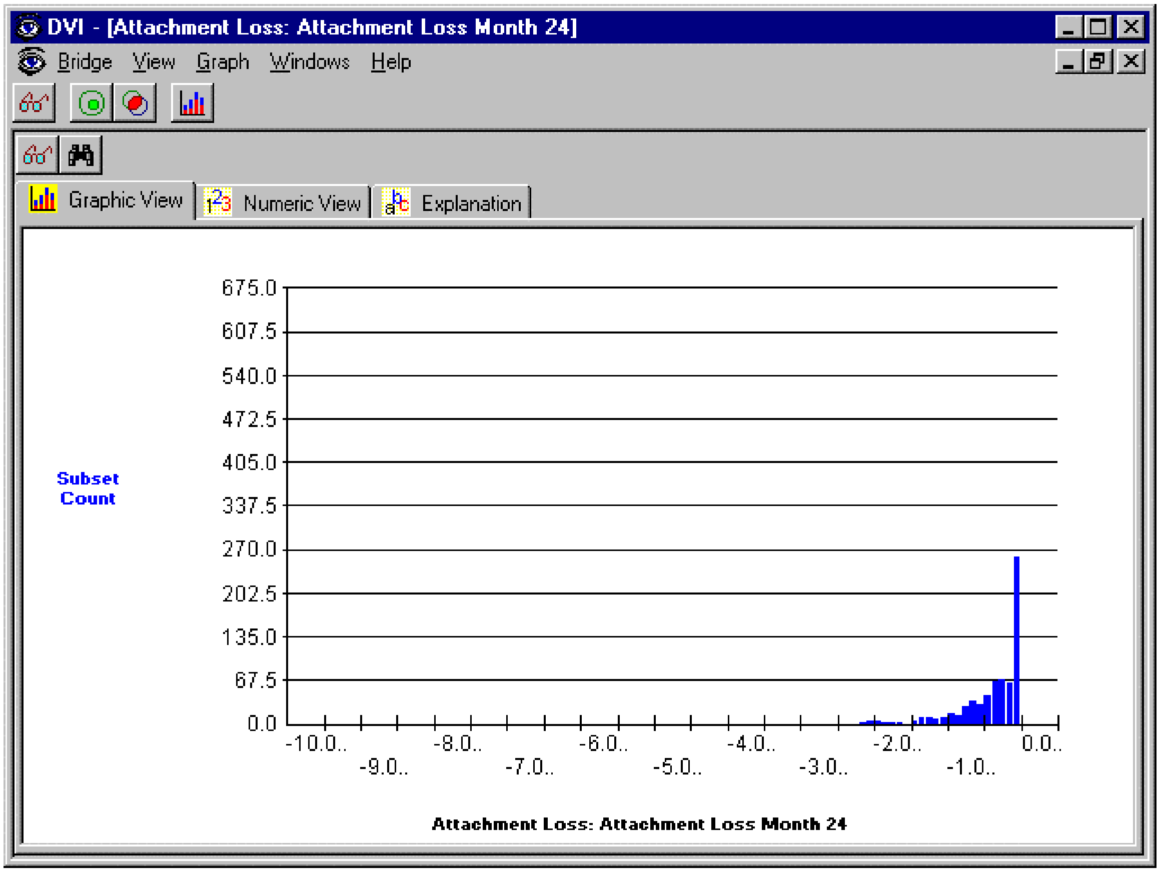


Figure 2: The Data Exploration Interface for a model of periodontal disease

## 4 CONCLUSIONS

In conclusion, the Experimental Design and Analysis Simulation Interface is a tool for individuals who wish to use their models to conduct studies of a complex domain. It supports data entry, experimental study generation, study analysis, and post hoc data mining and exploration of large, multidimensional data sets.

The EDASI is designed to support the design and analysis of a series of simulation runs in which the model is deterministic. Although the general concept of EDASI is relevant to stochastic models, the detailed implementation of an interface to a stochastic model would be substantially different. In deterministic models, the variation of a variable involves exploration of all logical values, while in stochastic models, the variation follows a probability distribution. The question in deterministic models is what effect does a systematic change in the value of a variable have on system behavior. On the other hand, the question with a stochastic model is how does the variance in an input variable influence the distribution of system behavior. Thus, though an interface for a stochastic model would still need the same three major components, including a Study Design Interface, a Simulation Results Database, and a Data Exploration Interface, each component would require significant alterations to support the differing needs of stochastic modeling. The Study Design Interface would need to allow for specifying distributions associated with input variables as well as support for determining replication/run length requirements. The data model of the Simulation Results Database would be substantially different in order to associate input and output distributions. Finally, the Data Exploration Interface would require a different visualization methodology as well as a variety of statistical analysis tools to support the necessary formal analyses.

The EDASI transforms simulation models into research programs. It makes it extremely simple for researchers to define the questions they wish to ask, design and run the appropriate studies that will generate data pertinent to answering the questions, evaluate the results, and design additional studies to refine and enhance their understanding of the system or process under investigation. Such support is needed to help make efficient and effective use of the models that are developed whenever the study of a large, complex system is conducted.

## REFERENCES

Benson, D. 1996. Simulation Modeling and Optimization Using ProModel. In *Proceedings of the 1996 Winter Simulation Conference*, eds. J.M. Charnes, D.M. Morrice, D.T. Brunner, and J.J. Swain, 447-452.

Jankauskas, L, and S. McLafferty. 1996. BestFit, Distribution Fitting Software by Palisade Corporation. In *Proceedings of the 1996 Winter Simulation Conference*, eds. J.M. Charnes, D.M. Morrice, D.T. Brunner, and J.J. Swain, 551-555.

Kelton, W.D. 1994. Analysis of Output Data. In *Proceedings of the 1994 Winter Simulation Conference*, eds. J.D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila, 62-68.

King, C.B. 1996. Taylor II Manufacturing Simulation Software. In *Proceedings of the 1996 Winter Simulation Conference*, eds. J.M. Charnes, D.M. Morrice, D.T. Brunner, and J.J. Swain, 569-573.

Krahl, D. 1996. Modeling with Extend™. In *Proceedings of the 1996 Winter Simulation Conference*, eds. J.M. Charnes, D.M. Morrice, D.T. Brunner, and J.J. Swain, 578-583.

Kuljis, J. 1996. HCI and Simulation Packages. In *Proceedings of the 1996 Winter Simulation Conference*, eds. J.M. Charnes, D.M. Morrice, D.T. Brunner, and J.J. Swain, 687-694.

Law, A.M. and M.G. McComas. 1996. ExpertFit: Total Support for Simulation Input Modeling. In *Proceedings of the 1996 Winter Simulation Conference*, eds. J.M. Charnes, D.M. Morrice, D.T. Brunner, and J.J. Swain, 588-593.

Leemis, L.M. 1994. Input Modeling. In *Proceedings of the 1994 Winter Simulation Conference*, eds. J.D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila, 55-61.

Markovitch, N.A. and D.M. Profozich. 1996. Arena® Software Tutorial. In *Proceedings of the 1996 Winter Simulation Conference*, eds. J.M. Charnes, D.M. Morrice, D.T. Brunner, and J.J. Swain, 437-440.

## AUTHOR BIOGRAPHIES

**L. TANDY HERREN** is a Staff Scientist for Medical Science Systems, Inc. She received a B.S. degree, an M.A. degree, and a Ph.D. in Psychology from Ohio State University. She also received an M.S. in computer science from Ohio State. Her current research interests include biological simulation to support medical research and development, and knowledge acquisition and representation methodologies in software development. She is a member of the American Association of Artificial Intelligence, the IEEE Computer Society, and the Society for Computer Simulation.

**PAMELA K. FINK** is Executive Director of Discovery and Technology Systems for Medical Science Systems, Inc. She received a B.A. degree in Mathematics and Philosophy from Eckerd College in 1989 and a Masters' and Ph.D. in Computer Science from Duke University in 1991 and 1993, respectively. Her research interests include issues in knowledge acquisition, representation, and utilization as they relate to problem solving and computer software implementation. Currently, she develops computer models of complex biological processes to support pharmaceutical and medical device research and development. She is a member of the American Association for Artificial Intelligence, the IEEE Computer Society, the Association for Computing Machinery, and the Society for Computer Simulation.

**CHRISTOPHER J. MOEHLE** is a Staff Scientist with Medical Science Systems, Inc. His background is in Linguistics and Computer Science, specializing in Artificial Intelligence. His current research interests include interface and database design, and data visualization.