Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

INTERACTIVE SIMULATION
(PANEL)

Chair

Charles R. Standridge

Department of Industrial Engineering
Florida A&M University / Florida
State University, P.O. Box 2175
Tallahassee, Florida 32307

Spotlight Presentation

Kenneth M. Matwiczak

Department of Systems Engineering
United States Military Academy
West Point, New York 10996-1779

Respondents

Deborah A. Davis

Systems Modeling Corporation
The Park Building
504 Beaver Street
Sewickley, Pennsylvania 15143

Kenneth J. Musselman

Pritsker Corporation
1305 Cumberland Avenue
P.O. Box 2413
West Lafayette, Indiana 47906

Daniel T. Brunner

Wolverine Software Corporation
4115 Annandale Road, Suite 200
Annandale, Virginia 22003-2500

INTERACTIVE SIMULATION: LET THE USER BEWARE!

KENNETH M. MATWICZAK

ABSTRACT

In recent years a significant amount of effort has gone into making simulation languages friendlier for the decision-maker and easier to use for the simulation analyst. One such area where these efforts are beginning to show results is in the area of interactive simulations, in which the user can observe and modify a simulation model as it is running. However, there are hidden perils in interpreting the results of the data output from an interactive simulation. This paper attempts to highlight some of these dangers, pointing out some statistical pitfalls with the traditional end-of-run post-simulation analysis. Some suggestions are made about the directions future research might take to provide more meaningful and statistically valid end-of-run results.

1. INTRODUCTION

A primary purpose of simulation modeling is to support the decision-making process. In the past, and to a great extent today, this meant that a person trained in the art and science of simulation had to be called upon to conduct the simulation study and subsequently present the interpreted results to the decision-maker.

In recent years, great strides have been made toward putting simulation in the hands of the primary user, the decision-maker. Interactive graphics, animation, comprehensive modeling environments, and interactive simulation languages have all played a role in the growth of simulation as a decision support tool.

Unfortunately, however, it does not seem that a comparable amount of work is being done in updating the statistical aspects of simulation to keep pace with the rapid changes in simulation languages and modeling environments. With few exceptions, most simulation output is still presented as a *batch* result such as the SLAM II Summary Report [Pritsker 1986] and the SIMAN Output Processor [Pegden 1986]. Means, variances, histograms, and plots are used to present a compilation of the simulation data over the entire simulation run. The impact of changes in the simulation model structure, or its parameters, at some point prior to the end of the run, is difficult to capture and even more difficult to present in a succinct form.

Simulations are used to conduct experiments, the results of which are used to support the decision-making process. As in any other experimental process, statistical analysis plays a key role from problem formulation to the resulting decision.

With respect to the simulation modeling process, statistical procedures are especially key in the following areas:

- Collection and modeling of input data.
- Process modeling (e.g. service times).
- Design of the simulation experiment.
- Model validation.
- Presentation of output data.
- Sensitivity analysis.
- Interpretation of results.
- Forecasting.
- Decision-making.

There has been much research conducted in each of these areas, some of which may not have been directed specifically towards the simulation environment, but the results and techniques of which can be applied to simulation. One area that has received some attention in recent years is simulation output data analysis [Law 1983]. Because of the stochastic nature of most simulations, this area is critical to understanding the dynamics of the simulation model and the relationship of system components, correctly interpreting the results of the simulation, and making statistically valid decisions.

The purpose of this paper is to raise the awareness of current, and potential, interactive simulation users, with respect to some of the statistical pitfalls in the interpretation and presentation of the output of interactive simulations. It attempts to raise some apparently neglected issues in this regard. After establishing a baseline definition of *interactive* simulation, those areas in which statistical analysis plays a role in the interpretation of simulation output, are briefly summarized. A simple example is used to demonstrate how traditional end-of-run statistics can be misleading for a simulation model modified in mid-run. Finally, some possible directions for future research are offered, which might address, and redress, this potentially dangerous trend.

2. INTERACTIVE SIMULATION

The word *interactive* has taken on many different meanings with regard to simulation. On the one hand, interactive simulations are those in which the user can build or modify a simulation model through some pre-processor program, which guides the user through the model-building process. These programs, such as ISIM [Hay and Crosbie 1984], use an interview approach, asking the user a series of questions, the responses to which define the model structure and parameters. Others, such as DYNANET [Matwiczak and Talavage 1980] use icons and symbols which the user assembles in a prescribed manner to represent the system under study. The end result in either case is the same. The interactive program assembles a model in some simulation language, which is then executed to termination. The user cannot interact with the model while it is running without stopping and restarting the simulation.

Another approach to interactive simulation is via animation and interactive graphics. Some simulation packages present the model in the form of an animation, either as it is executing, as does SAINT PLUS [Micro Analysis and Design 1989], or as a post-process, such as in TESS [Standridge and Pritsker 1987]. SAINT PLUS allows the user to observe the dynamics of various model parameters by means of histograms, meters, etc. Thus, the user realizes an appreciation for the dynamics of the simulated system and can visually identify potential problems, such as choke points and blocking, as the simulation is running. Some of these same simulation packages permit the user to stop the model as it is executing, change one or more of the model parameters, and continue the simulation run, as described by Gilman [1985].

Finally, a third type of interactive simulation is being manifested as a result of the work in Artificial Intelligence (AI). The application of AI techniques and object-oriented programming is providing for a greater scope of interaction.

An objective of marrying AI and simulation is to enable the simulation model and database to learn from each simulation run, user query, etc. This adds a new dimension to the term *interaction*, since now the simulation is not only interacting with the user, it is also interacting with its own history via the model information base.

For purposes of this paper, *interactive* simulation is defined to be any program in which the user can suspend execution of the model, modify one or more parameters, including the model structure, and resume model execution. In this context, the interaction could be via graphics or some other form of interrupt. The simulation eventually terminates at some predefined time or upon meeting some specified condition.

The benefits of interactive simulation, to both the simulation analyst and the user, are many and may include:

- Greater control of the simulation run.
- Easier model verification.
- A better sense of the system dynamics.
- Enhanced presentation.

However, there are potential dangers hidden in the interpretation of interactive simulation results which, if left unheeded, could result in faulty decision-making based on erroneous information. Ultimately, decision-makers' confidence in simulation modeling as a valid decision tool may be shattered.

3. STATISTICS AND OUTPUT ANALYSIS - A QUICK SUMMARY

Many approaches to the analysis of simulation output data have been offered over the years, depending on how the data is characterized by the analyst. For example, see Fishman [1978], Law [1983], Kelton [1985], or Kleijnen [1987].

Traditional parametric statistics look primarily at the data as the results of many experimental observations, either independent or correlated, and attempt to make inferences based on parameters of the data, such as expected values and standard deviations. The time series approach to output data analysis considers the data to be a series of autocorrelated values and attempts to summarize the simulation results as such. Still other approaches to analyzing simulation output have been suggested, ranging from factor screening techniques [Smith and Mauro 1982; Bauer 1985] to percentile estimations [Seila 1982] and other non-parametric approaches [Friedman and Friedman 1985], to name only a few.

However, a common thread among all these approaches is that they use a *batch* approach to the presentation and analysis of simulation output data. That is, the data is analyzed as a batch of observations obtained over the entire simulation run, under system parameters and conditions which do not change during the course of the simulation run.

The advent of interactive simulation, which permits changes to model structure and parameters during execution, complicates the analysis of the end-of-run output data, and in many cases invalidates the statistics used to describe the collected data. The data accumulated at the end of a simulation run is no longer based on the same assumptions, interactions, and distributions established at the beginning of the run, nor can it be characterized by the assumptions and parameters in effect at the end of the simulation. Changing the simulation model parameters during execution gives additional meaning to the word *transient* with respect simulation output data. Assumptions about stationarity, normality, etc., become even less relevant, but no less important, in interactive simulation.

4. AN EXAMPLE

To illustrate the difficulty of correctly interpreting end-of-run results of a simulation which has been changed *on-the-fly*, four cases of a simple M/M/1 queue are simulated in SLAM II. For each case, the customer interarrival times are exponentially distributed with a mean of 1.0 time units. Each simulation is run until 2000 customers have been serviced. In the first and second cases, the customer service times are exponentially distributed with a mean of 1.0 and 0.5 time units, respectively. In case 3, the first 1000 customers are assigned and exponential service time with a mean of 1.0 time units. The second 1000 customers have service times that are exponentially distributed with a mean of 0.5 time units. In the fourth case, the assigned service times were reversed from case 3; that is, the first 1000 customers had mean service times of 0.5 time units, while the second 1000 customers were serviced at a rate of 1 per time unit.

Cases 1 and 2 were executed to establish a frame of reference for the primary demonstration cases, 3 and 4. Cases 3 and 4 represent an interactive simulation which is interrupted after 1000 parts have been serviced, the future service times distribution is changed, and the simulation allowed to execute to completion. Table 1 summarizes the differences between the exponentially distributed service times for each case.

In all four cases, common random number streams were used to assign inter-arrival times and service times, so that the order of customer arrival and assignment of service times was consistent between all the cases. Selected results of each model run, as extracted from the SLAM II Summary Report, are summarized in Table 2.

Without statistical analysis, one can see that the results of cases 3 and 4, as measured by key parameters of a queuing system, are significantly different from each other. If either case 3 or case 4 had been run as an isolated experiment using

Table 1. M/M/1 Mean Service Times

	Case 1	Case 2	Case 3	Case 4
Mean	1.0	0.5	1st 1000 1.0 1st 1000 0.5	1st 1000 0.5 2nd 1000 1.0
Change In Run	No	No	Yes	Yes

an interactive simulation language, the inexperienced decision-maker, or inattentive simulation analyst, might be tempted to look at the end-of-run results, assume that the simulations ran long enough to achieve steady-state, and accept the results within some confidence interval.

It is a simple matter to calculate the expected values and other descriptive statistics of various measures of performance for cases 1 and 2, assuming steady-state conditions. However, the analysis of cases 3 and 4 is not quite as straightforward. Several questions are raised, the answers to which, if not addressed properly, could adversely affect the interpretation of the summary statistics and result in subsequent decisions based on the faulty analysis.

A most obvious question would be, "Why are the performance measures, such as queue length and average time in system, so different between cases 3 and 4?" Both runs simulated 2000 customers, half of them serviced at a rate of 1 per time unit and half at 2 per time unit. The answer lies in the answers to subsequent questions, which themselves depend on the amount of information presented to the analyst in the output report.

In order to properly interpret the average queue length for those cases in which mean service times changed during the model execution, one would need to know the status of the queue when the service times actually changed. The queue length, at the time of the change, effectively becomes an arbitrary start-up condition. Many simulation models are started with queues initially empty and idle, and are then run for some period of simulated time to overcome these start-up conditions, before recording data for analysis. In interactive simulation, the length of time needed to achieve steady-state after changing a model parameter becomes a random variable, which depends on the status of the system at the time of the change. Thus, in the interactive simulation, there are more than one set of *pseudo*-start-up conditions to account for in the end-of-run statistics.

Cases 3 and 4 demonstrate another question raised in interpreting the output from interactive simulations. "When, in simulation time, did the change occur?" the answer to this question is easily captured as a simulation event time, but what is the impact on the summary statistics presented at the end of the run? In the M/M/1 system, it is reasonably safe to assume that the change in service time occurred earlier in simulation time in case 4 than in case 3. Why would this be important to us? Many of the summary statistics presented at the end of a simulation run are *time-dependent*. That is, their expected value depends on how long the variable being investigated was in a certain state over the course of the run. Average queue length is a good example of this. Effectively, the longer a variable is in a given state, the more weight it carries in determining the overall average value of that variable. Queue lengths tended to be shorter, for shorter amounts of time early in case 4, while they tended to be longer for longer periods initially in case 3. The result of this arbitrary weighting is to bias the summary statistics in some manner, unless the parameter change time is accounted for in calculating summary statistics.

The example presented here described a simple M/M/1 queue, which in a simulation study would be only a small component of a vastly more complex system. What would be the impact on the other components of the system of a simple change? Summary statistics based on observations over the entire simulation run provide no way to measure or highlight this *ripple effect*. What about the effect of changing more than one model parameter, or even the model structure, in the course of the simulation run? This presents a potentially huge data capture problem, and an even more complex analysis problem. This is especially true if the analyst is to rely solely on the post-run summary statistics as a basis for analyzing system performance. Any attempt to do quantitative sensitivity analysis of the system becomes more difficult since much of the important information is now confounded by the impact of changes directly on the parameter of interest and indirectly via the ripple effect.

It might be useful and convenient for the analyst to simple *average* the changed parameters over the length of the simulation and use this average in calculating the summary statistics, such as taking a weighted average of the two service time distributions used in the example. Since simulation models represent a stochastic process, this would be neither easy nor desirable. It is not easy for the reasons cited above with respect to start-up conditions and time-weighted averages. It is not desirable because the order in which the changes to the model are made also becomes important, as evidenced by the differences between case 3 and case 4 results. By predetermining the order of the changes, the user is artificially inducing a determinism and correlations into a random process, which must also be taken into account in any analysis.

5. SO WHAT?

If the eventual goal of ongoing simulation research efforts is to put simulation into the hands of the decision-maker, then it is important that we, as simulation developers, analysts, and researchers, ensure that the simulation tools being used are relatively easy to use, understandable, and, above all, provide the user with (statistically) valid information and results.

Table 2. Summary of M/M/1 Queue Simulations

CASE #	AVG Q-LENGTH	MAX Q-LENGTH	AVG WAIT TIME	SRVR UTILIZATION	AVG TIME IN SYS
1	17.75	45	17.39	0.98	18.4
2	0.403	7	0.394	0.494	0.89
3	10.02	39	9.78	0.742	10.5
4	7.88	45	7.72	0.731	8.4

Interactive simulation, and the marriage of AI and simulation, are making great strides toward more understandable simulations that are easier to use. However, we also need to ensure that the results presented to the user at the end of the simulation, or at any point in the run, are meaningful and statistically valid if far-reaching decisions are to be based on the simulation results.

One possible simple solution might lie in modifying the type of summary information presented at the end of a simulation run. This information might include the times and types of changes made during the course of the model run. Presenting these simple events and data may at least be enough to highlight the fact that the summary statistics need special interpretation. It would then be up to the user to determine the impact of the changes.

Another potential solution might be to analyze the simulation data as the simulation is running, accounting for parameter changes as they occur. Most simulation languages already do this by virtue of their data collection and storage techniques and statistical calculations. However, these routines need to be modified in several ways. One modification might be to change how the summary statistics are presented, to include presenting pre- and post-change statistics. This may be extremely cumbersome and result in lengthy summary reports with numerous bits and pieces of data. Most likely, it would also require increased data storage capability. Again, it would be up to the user to account for an *pseudo* start-up conditions, determine if, and when, steady-state had been achieved, and decipher the information that is presented.

A second approach to modifying existing programs may be to program the statistical routines in the simulation language to automatically account for induced correlations or to make allowances for the changed parameters. This infers the ability to determine, at the time of the change, how far *down time* the change ripples, and what the statistical impact is on subsequent parameters. This would possibly require a significant increase in data storage and retrieval capability to retain the necessary historical information. The logic involved in implementing this analysis would not be trivial, resulting in significant additions to already large problems, or at least program modifications that may not yet be cost effective. This alone implies a need to incorporate AI techniques in the data analysis and presentation routines.

The final solution may lie in combining simulation and statistics with AI techniques. Again, not a simple task. There has been some success in applying AI to simulation analysis, for example Reddy, et al. [1986] and Matwiczak [1990]. The use of a "physicist" object in ROSS [MacArthur et al. 1985] is an example of how the user can tailor the simulation program to calculate only those statistics of interest. Within this object, the programmer can encode the logic need to account for transient analysis or interactive model changes. But this, too, still puts the burden on the simulation user to determine what logic is need for each simulation and how to account for any ripple effects. Although this is a step in the right direction, it still is some distance from putting simulation directly into the decision-makers' hands.

Another effort in the application of AI techniques to the analysis of simulation results is in Reddy et al. [1986]. Their Knowledge Based Simulation system (KBS) is an object-oriented simulation system that uses AI logic techniques to achieve user-specified goals. Already programmed into the logic are modules that will rate a scenario on how well it achieves specified goals, perform a causal path analysis, and/or perform a user-specified rule-based diagnosis. Thus, KBS seems to point a way to including transient analysis and reporting the impact of dynamic changes in interactive simulations. However, work still needs to be done to ensure that the simulation results summary is meaningful to the decision-maker, while remaining statistically valid. The potential still exists to make the analysis of simulation output data more transparent to the user and present the results in more *useable* form.

Parametric statistical analysis is proving difficult to implement with AI techniques because of the amount of *human* interaction often involved [Gale 1986]. However, Exploratory Data Analysis (EDA) [Tukey 1977] and non-parametric techniques, when coupled with AI techniques, might have applications in simulation output data analysis. Improvements in computer processing capabilities and storage technology can only facilitate solutions to the problem of transient data analysis of interactive simulations.

6. CONCLUDING REMARKS

It would seem counter-productive and contrary to the ultimate goal of placing valid, usable simulation tools directly in the hands of the decision-maker if we required the simulation program user to make special allowances in his or her model, write additional code, or artificially limit the experimental changes to the model. The user should not be required to expend this added effort just to capture all the necessary information about a model changed "on-the-fly", or to conduct a statistically valid analysis and data interpretation. There should be a way to relieve the user of these burdens and yet instill confidence that the results presented at the end of the simulation run are meaningful, complete, and statistically valid.

As an area for future research, the presentation and analysis of interactive simulation output appears to be fertile ground. The research being conducted in the application of Artificial Intelligence techniques to simulation modeling and statistical analysis provides a vehicle for potential solutions. Until then, let the user beware!

REFERENCES

Bauer, P., B. Kochar, and J.J. Talavage (1985), "Simulation Model Decomposition by Factor Analysis," In *Proceedings of the 1985 Winter Simulation Conference*, SCS, San Francisco, CA, 185-188.

Fishman, G.S. (1978), *Principles of Discrete Event Simulation*, John Wiley and Sons, New York, NY.

Friedman, L.W. and H.H. Friedman (1985), "The Probability Distribution as a Performance Criterion When Comparing Alternative Systems," *Simulation 45*, 5, 262-264.

Gale, W., Ed. (1986), *Artificial Intelligence and Statistics*, Addison-Wesley Publishing Company, Reading, MA.

Gilman, A. (1985), "Interactive Control of the Model: A Natural Companion to Animated Simulation Graphics," In *Proceedings of the 1985 Winter Simulation Conference*, SCS, San Francisco, CA, 196-198.

Hay, J.L. and R.E. Crosbie (1984), "ISIM - A Simulation Language for Microprocessors," *Simulation 43*, 3, 133-136.

Kelton, W.D. (1985), "Analysis of Simulation Output Data," In *Proceedings of the 1985 Winter Simulation Conference*, SCS, San Francisco, CA, 33-34.

Kleijnen, J.P.C. (1987), *Statistical Tools for Simulation Practitioners*, Marcel-Dekker, New York, NY.

Law, A.M. (1983), "Statistical Analysis of Simulation Output Data," *Operations Research 31*, 6, 983-1029.

MacArthur, D., P. Klahr, and S. Narain (1985), "The ROSS Language Manual," Rand Note #N-1854-1-AF, Rand Corporation, Santa Monica, CA.

Matwiczak, K.M. (1990), "AMOS - Automated Multiple-Response Optimum-Seeker for Simulations," Ph.D. Dissertation, Department of Industrial Engineering, Texas A&M University, College Station, TX.

Matwiczak, K.M. and J.J. Talavage (1980), "DYNANET - Network Modeling for Systems Dynamics," In *Proceedings of the 1980 Summer Simulation Conference*, SCS, Seattle, WA, 173-178.

Micro Analysis and Design (1989), *SAINT PLUS*, Demonstration Software, Boulder, CO.

Pegden, C.D. (1986), *Introduction to SIMAN*, Systems Modeling Corporation, State College, PA.

Pritsker, A.A.B. (1986), *Introduction to Simulation and SLAM II*, John Wiley and Sons, New York, NY.

Pritsker Corporation (1989), *SLAMSYSTEM: Total Simulation Project Support,"* Demonstration Software and User Guide, Indianapolis, IN.
Reddy, Y.V.R., M.S. Fox, N. Husain, and M. McRoberts, "The Knowledge-Based Simulation System," *IEEE Software 3*, 2, 26-37.
Seila, A.F. (1982), "Estimation of Percentiles in Discrete Event Simulation," *Simulation 39*, 6, 193-200.
Smith, D.E. and C.A. Mauro (1982), "Factor Screening in Computer Simulation," *Simulation 38*, 2, 49-54.
Standridge, C.R. and A.A.B. Pritsker (1987), *TESS - The Extended Simulation Support System*, John Wiley and Sons, New York, NY.
Tukey, J.W. (1977), *Exploratory Data Analysis*, Addison-Wesley, Reading, MA.

DEBORAH A. DAVIS

The use of interactive simulation, in which a person (modeler or other user of the simulation model) is able to examine and change simulation parameters during an executing simulation run, has been a great boost to the growing popularity of computer simulation. However, "let the user beware" should be stamped across the logo of simulation software, as there is strong potential for misuse of this capability. Also, simulation software vendors need to investigate development of interactive software to support other tasks encompassed by simulation studies.

Interactive simulation is a great aid to the modeler in debugging and validating a model and in training personnel who will operate a system, especially when concurrent graphical animation immediately displays the effects of interactive model changes. Short, interactive runs also may be helpful in paring down the number of system alternatives to be fully evaluated.

However, interactive simulation runs also have inherent dangers, especially regarding the interpretation of results of replications in which the system was modified during the final experimentation phase of a simulation project. In particular, once a model has been deemed complete (i.e., bug-free and appropriate to its purpose), the temptation to continue to play with system settings and to attempt to analyze the results of these runs is aggravated by the ease with which these changes can be made.

During final experimentation and analysis, interactive changes to simulation models should not be made. Once a model and its supporting data have been verified, making interactive changes to system parameters runs the risk of introducing new bugs by specifying the wrong information or making changes at the wrong time. Also, any experiment should be reproducible with the software used to perform the initial runs; interactive changes introduce a new variable in the ability to reuse a simulation model to verify past analysis or to analyze new alternatives.

Finally, while the term "Interactive Simulation" usually refers to interaction with a simulation as it executes, I would like to see both the attitude toward simulation studies and the capabilities built into simulation software take a broader approach to user interaction.

Many simulation packages provide interactive model building and model execution, but provide little support for other aspects of simulation projects. In the model preparation phase, there is little commercial software support. Interactive software to assist in formulation of a conceptual model, preliminary experimental design, and preparation of input data would provide a great service to simulation modelers. During model development and execution, many simulation languages provide interactive software with the characteristics previously described, but at the back end of simulation projects, there is again little assistance for detailed experimental design, run-length planning, and analysis of results.

If new approaches and software are developed for supporting the complete simulation process, many more successful simulation studies will be carried out. This is the charter in "Interactive Simulation" that faces simulation software developers.

KENNETH J. MUSSELMAN

Simulation is a complex decision support process that involves a range of activities. These activities include:

- Problem formation,
- Model conceptualization,
- Data collection,
- Model building,
- Verification,
- Validation,
- Analysis,
- Documentation, and
- Implementation.

All of these activities require interaction. In recognition of this need, simulation support systems, such as SLAM II with TESS and SLAMSYSTEM, have been developed. These support systems have made significant use of interactive technology. Data can now be statistically as well as visually characterized, and models can be graphically developed, visually verified and validated, concurrently animated, interactively analyzed, and creatively documented.

The simulation process has been improved through the use of these simulation support systems. Creative ideas have been produced that would have otherwise never been generated. Errors of omission and commission have been identified that would have otherwise gone undetected. Profound and meaningful system experiences have been felt that would have otherwise been missed. Significant and complicated cause-and-effect relationships have been discovered that would have otherwise gone unnoticed. Results have been implemented with more certainty than would have otherwise been possible.

With these benefits, however, come reasons for concern. Most of this concern is rightfully centered around execution and analysis. Inconsistent decisions, myopic judgements, over reaction to transient behavior, over emphasis of a single run, over promotion of subjective based analyses, nullification of standard statistical procedures, and inability to implement are some of the reasons behind this concern.

This is not to say that interactive modeling and analysis should be abolished. On the contrary, it should be judiciously used and encouraged. Simulation has gained added strength through its use. The challenge is to learn how and when to use this medium to better the decision making process.

DANIEL T. BRUNNER

Traditionally, simulation modeling has been viewed as a technique to be used in support of the decision-making process. New tools for interactive simulation, including both interactive model builders and interactive experimentation environments, allow modeling to become an integrated part of the design and decision-making process instead of a detached analysis tool. Is this a dangerous trend?

The traditional view of simulation projects includes phrases like "gather the data" and "present the results." This assumes that the analysts performing the simulation "study" operate in a cleanly detached fashion from the system designers and decision-makers. Thanks to interactive tools, which are generally easier to learn than simulation languages, this distinction is blurring. Analysts are working more closely with design groups. Sometimes the designers themselves become the simulation "analysts." (We use the term "designers" broadly, to include anyone who makes design recommendations or decisions about the current or future operation of a system.)

Unfortunately, the designers are often completely untrained in simulation. In fact, the attraction to interactive tools may arise because no one in the group knows how to use traditional simulation software. Sometimes no one has even had a single simulation class in school! In the case of interactive simulation experimentation, designers may not care about the statistical implications of changing a model in mid-experiment because they may not be schooled in statistical analysis of output in the first place. They are more interested in looking at visual queues or point estimates of throughput or utilization and in performing quick (yet sometimes effective) simulation experiments by trial-and-error.

Clearly this represents an erosion of the effectiveness of simulation as it has been practiced traditionally. "Results," if examined at all, will be somewhat less reliable. The designer-turned-analyst might not possess the knowledge to perform a detailed simulation study, or the experience to recognize when one is required. Also, the important operational knowledge gained from the step-by-step process of doing a detailed simulation study -- arguably more important than final quantitative results in any simulation project -- is diminished, although this effect may be offset by the increased involvement of the designers.

Given this scenario, the future is hardly bleak for simulation. For one thing, traditional simulation is far from dead. Managers are increasingly aware of the benefits of performing detailed simulation studies that include accurate modeling of process flows and control logic, and can recognize situations where interactive tools might fall short. On the interactive side, it is the simulation profession's responsibility to recognize the emerging designer-oriented user community and continue to promote, enhance, and successfully apply simulation within it. As an example, the profession should seek to expand university course offerings and industrial training in discrete event simulation in design-related fields such as mechanical and electrical engineering.

Can the profession help overcome the often-cited problem of invalid statistics in an interactive experimentation environment? The penalty imposed may not be as severe as once thought. An uneducated user will be unconcerned about statistics from the start, and an educated one ought to know the difference between interactive "fiddling" -- which has its place -- and formal experimentation. Tool builders could still assist both types by suppressing all automatic statistical output following any interactive change. There may not be much difference between unused statistics, bad statistics, and no statistics, but most would agree that bad statistics are the worst of the lot.

Interactive simulation presents problems in many other areas as well. Unless the interactively built model is properly formulated (a problem area), is based on good input data (another problem area), and is properly verified and validated (this will probably be done partly or mostly through animation, which is effective but not rigorous), then whatever experimentation is performed will be of little value and might well be dangerously misleading. Through education and better software tools, these problems can be addressed. A little simulation knowledge goes a long way.