Proceedings of the 1988 Winter Simulation Conference
M. Abrams, P. Haigh, and J. Comfort (eds.)

Introduction to simulation

Arne Thesen
Department of Industrial Engineering
University of Wisconsin-Madison
Madison, WI 53706

Laurel Travis
Management Department
University of Wisconsin-Whitewater
Whitewater, WI 53190

I. INTRODUCTION

In today's competitive business climate, careful planning and analysis of alternative strategies and designs is essential. In an effort to derive maximum benefit from available resources, engineers and business planners have therefore made performance analysis an important part of their planning activities. Among the many available performance analysis tools, simulation in particular, has dramatically increased in popularity due to its broad range of applicability. Here we will give an overview of simulation modeling and analysis from the perspective of somebody wanting to use simulation *as a decision aid.* At the end of this tutorial, you should have a general understanding of simulation and an understanding of its applicability to your situation.

The type of simulations discussed here are used to develop an understanding of the performance of a system over time. For example, we may want to understand how scheduling rules affect the mean time in the system for jobs in a Flexible Manufacturing System (FMS). To arrive at this understanding, we would build computer models showing how parts flow through the FMS under the different rules. The model would use random variables to replicate variability in quantities such as the service times for the individual parts. Then we would run the models (i.e., operate the simulated FMS), accumulating the length-of-stay for individual parts as they leave. This data would then be used as a basis for comparing the rules and selecting one for implementation.

A. Questions Answered by Simulation

For simulation to be effective, it must be focused on some previously defined problem (otherwise we do not know that to include in the model and what information to collect). Using simulation before a specific problem is articulated may lead to a large number of simulation runs that use inappropriately designed models, and accordingly produce little or no information of value.

Different questions are asked at different stages of a study, and they are answered by models with different levels of detail. For example, questions about overall plant capacity are frequently asked early in the project when few details about the design are available and fairly rough answers may suffice. In this case a simple model is appropriate. On the other hand, questions about the efficiency of different scheduling rules in an automated manufacturing line can only be answered when the detailed design of the system is finalized and precise speeds, capacities and part routings are known. In this case a detailed model is required and a fairly sophisticated analysis of the simulation output is called for. Some typical questions that can be answered through simulation are given in Table 1.

```
Do I have sufficient capacity?

Does my design work?
     Does the food arrive before it is
     cold?
     Does the elevator have sufficient
     capacity?

Which is the better alternative?
     MRP or just-in-time?

Where is the bottleneck?
     Is capacity restricted by the
     grinder or the mill?

Fine tuning of system
     How many grinders?
     What is the optimal buffer capacity
```

Table 1: Some questions that can be answered by simulation.

While our subject is simulation, we should point out that many of the questions that arise at the early stages in a design project may be answered with sufficient precision using simpler models and specialized computer packages. For example a simple Lotus 123 model may be used to balance production lines such that each station works at approximately the same rate. Also, quick modeling tools such as Manuplan (Suri and Tomsicek (1988)) can be used to estimate the average time to get a product through a manufacturing cell. The advantage of these approaches is that they are easy to use, and they provide fast answers to "what-if" type questions. Several weeks may be required if simulation analysis is used. The disadvantage of these simpler analytic models is that their models are often inexact representations of the phenomena of interest, and even the most expertly formulated model may be off by 10-20% or more. However, such answers are often satisfactory in the planning stage.

Application area	Performance Measure
Air traffic control	Delays
Bank teller scheduling	Waiting times
Electric car	Battery usage
Computer networks	Delays
Scheduling rules	Throughput/day
Harbor management	Delays
Location of Fire stations	Response times
Social systems of wasps	Nest building

Table 2: Some recent applications of simulation

B. Application Areas

The military services were among the first to use simulation analysis, and simulations ranging from evaluations of maintenance policies to large scale war games are routinely used to guide defence policy. Simulation is rapidly growing in popularity and examples of successful simulation applications are found in a surprisingly eclectic range of fields (Table 2).

The most recent growth in simulation applications has been in the manufacturing area. Almost all major new construction projects are accompanied by some sort of simulation analysis, and many planned changes of manufacturing processes benefit from the insights gained from simulation.

C. Some Common Pitfalls.

Simulation analysis is not without drawbacks. *First,* the quality of the analysis depends on the quality of the model; model building is an art. *Second,* it is often difficult to determine if an observation made during a simulation run is due to a significant underlying relationship in the system being modeled or due to the built-in randomness of the run; simulation results are hard to interpret. *Finally,* simulation analysis is usually a time-consuming and expensive process, and an adequate analysis may not be feasible within the time available; analytic methods may be better for "quick and dirty" estimates.

1. Failure to state a clear objective
2. Failure to frame an answerable question
3. Using simulation when a simple analytic model would suffice
4. Analysis at an inappropriate moment
5. Inappropriate level of complexity
6. Bad assumptions in model
7. Poor output analysis
8. Budget overruns

Table 3: Some pitfalls to avoid

Some common pitfalls are listed in Table 3. Perhaps the most important of these is the failure to clearly state the objective of the project before it is undertaken, and to be guided by this purpose throughout the life of the project.

II. BUILDING SIMULATION MODELS

When beginning a simulation, it is often tempting to build a model of phenomena that are easily observed and understood. For example, if we want to understand the effect of reliability on machine utilization, we may be tempted to describe in detail how the machine works. However, this may be inappropriate, as the level of detail and time resolution required to describe machine operation is different from that of describing machine failure. It is therefore a good idea as a first step in the modeling process to develop the simplest possible model that provides the necessary information. Starting with such a rough model enables the modeler to describe some of the important relationships in the system without excessive detail. The insights gained from this simple model can then be used to aid in the effective development of a more detailed model.

A. The Role of the Simulation Model

To illustrate how simulation models are used as decision aids, consider the simple production process shown in Figure 1. We want to maximize machine utilization by specifying appropriate buffer sizes between the different work stations. Large buffers tend to result in higher utilization as no machine is starved for work. However, there is a point beyond which additional buffers add to overall cost without significantly improving utilization. To determine the effect of different buffer sizes, the analyst would therefore like to have a model such as the one shown in Figure 2.

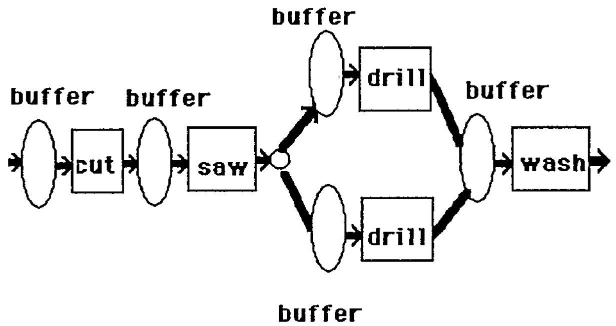

Figure 1: Simple production process

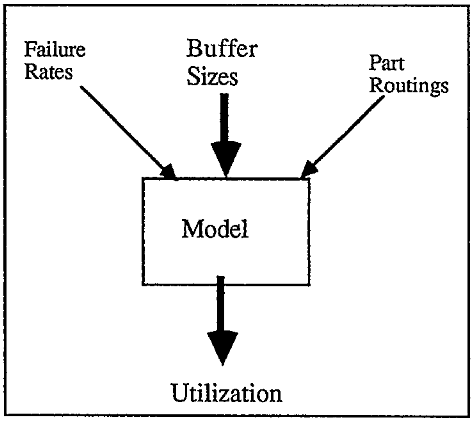

Figure 2: Model for analysis of effect of buffer size on utilization.

Ideally we would have a specific analytic model such as:

$$u = 1.54 - e^{-B}$$

where **u** = utilization, and
B = buffer capacity.

However, since we do not usually have prior knowledge of any such relationship, we use a simulation model that mimics the dynamic operation of the facility. As this model is running, we collect data about machine utilization, and at the end of the run the computer produces a report giving the average utilization for the run. So the simulation gives information about utilization for a given buffer size, but it does not compute optimal buffer size, or even give us a general rule for describing utilization as a function of buffer size. *In other words, the model does not explicitly describe the relationships that we want, rather, it describes how the system operates.* It is the job of the analyst to determine how the information needed for decision making can be obtained from the model.

B. Building Blocks of Simulation

Our ability to develop simulation models of a wide range of different phenomena is due to the fairly universal nature of the building blocks on which the models are based. In particular, the representation of dynamic behavior and the use of random variables are fundamental to all simulations. These two concepts are discussed in this section.

1. Modeling Elementary Random Processes

Regardless of the simulation package used, our goal is to replicate real life phenomena in the computer. For example, if we are studying the effect of different repair policies we need to generate intervals between machine breakdowns that model the intervals observed in the factory. Instead of carrying out a detailed analysis of the state of each machine in the system so that we can predict the exact time of breakdowns, we use random variables to represent the *pattern* of breakdowns regardless of cause. The time of any one simulated breakdown will be different from what we observe in real life but the long range pattern of breakdowns should be indistinguishable from the real life process.

Phenomena	Example	Typical Distribution
Choice	Tossing a coin Turning right or left	Bernoulli
Frequency	Breakdowns per hour Takeoffs per day	Poisson
Quantity	Age of a victim Actual weight of product	Normal
Interval	Time between breakdowns Time between arrivals	Exponential
Durations	Time to complete a task Time to repair a lathe	Erlang Gamma

Table 4: Phenomena frequently described by random variables.

Most simulation models use random variables this way to compensate for our lack of detailed knowledge of what is going to happen at any one instance in a real life process. Phenomena modeled this way include *choices, quantities, frequencies, intervals* and *durations*. Probability distributions describing these phenomena are readily available. Some frequently modeled random processes and their recommended distributions are given in Table 4.

We use the phrase *fitting of distributions to empirical data* to describe the process of finding a probability distribution with the property that random observations drawn from it are indistinguishable from empirical observations of a phenomenon of interest. The procedure for fitting data to distributions typically includes goodness-of-fit tests such as the Chi-square or Kolmogorov-Smirnov test and techniques for parameter estimation. Several software systems are available for this analysis (Thesen (1985), and Law and Vincent(1987)).

In addition, we are frequently asked to simulate situations about which we have limited knowledge -- we cannot fit a distribution to the data when there is no data. For example, we may be asked to evaluate the effect of different scheduling policies in a not-yet-constructed production system. In the absence of information other than the desired mean service time, it is convenient to specify a service time distribution such as a uniform or exponential. These choices are attractive as they do not need any additional information. However, exponential distributions tend to over-estimate the variability of a process, and uniform distributions tend to under-estimate variability. In either case the resulting output will be misleading. Selecting appropriate distributions in these situations requires a great deal of experience and judgement, or the gathering of additional information.

2. Describing Dynamic Behavior

Almost all simulation models are run by tracing the sequence of events that change the state of the system of interest over time. However, since we do not normally think of models of systems in terms of events and state changes, the modeler usually uses a "friendlier" representation. The computer then translates this to an event oriented approach to actually run the model. One such "friendly" approach, the transaction flow approach will be discussed here.

Many simulations describe how *transactions* flow through a *block diagram* representing a system of interest. For example, transactions may represent subassemblies and the block diagram may represent the flow of these subassemblies through an assembly process. By using standard blocks to describe what happens to transactions, these languages are able to represent the behavior of a wide range of different systems using a limited number of building blocks.

The first block (or statement) in a model generates transactions. For example, transactions representing individual customers in a waiting line or *queuing* system might be generated at random time intervals. Each transaction immediately flows through the diagram until it hits some obstacle that causes it to be delayed. Eventually, conditions change and the delayed transaction is allowed to move again. Two important mechanisms that cause the flow of transactions to be impeded are:

Explicitly specified delays -	the transaction waits in a block while being served, and
Blocking -	the transaction is refused entry to the next block.

Blocking usually occurs when a transaction wants to use a *resource* that currently is unavailable. For example, the transaction may want to receive the attention of a server that currently is busy serving somebody else. A running model may therefore contain a large number of transactions simultaneously.

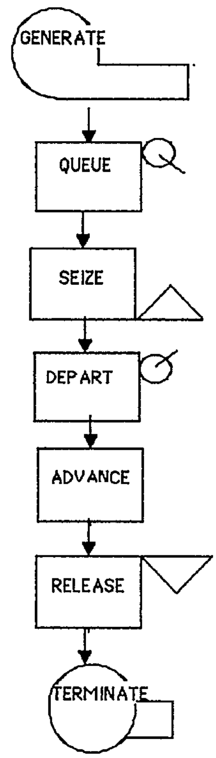

Figure 3: Graphical representation of a single server queuing model in GPSS.

A feature of the transaction-flow approach to model representation, is that resources are not always explicitly shown in the model. Instead, we show how the resource and the transaction interact. Accordingly, languages provide blocks to request the use of a resource and blocks to release control of resources. A diagram of a single server queuing model written in GPSS/PC is shown in Figure 3. Instead of explicitly showing the server, a request for service is represented by the SEIZE block and the release of the server is represented by the RELEASE block.

Identical models would result if other dialects of GPSS such as GPSS/H (Henriksen and Crain (1983)) were used. Very similar models would result if any of the other major languages such as SIMSCRIPT (Markowitz et al (1987)), SIMAN (Pegden(1986)), and SLAM(Pritsker(1986) were used. Other approaches to model representation are discussed in the following section.

C. Commonly Used Modeling Packages.

While it often easy to describe a situation to other humans, (for example by using graphs and/or analogies), it can be exceptionally difficult to capture its detail on a computer. We therefore try, whenever possible, to design models that draw upon previously developed models and programs.

The more choices that the model builder must make, the more difficult does the modeling process become. On one extreme, with special purpose simulation systems such as Sandie (Thesen (1986)) and StarCell (Steudel (1987)), users are shown a model *template* giving model structure and default parameters. For example the model we saw in Figure 3 is a standard feature in Sandie. After changing a few parameter values, it is ready to run. Although this is convenient if the pre-programmed model accurately reflects the problem at hand, a disadvantage is the inability to describe special circumstances.

Users of many of the emerging interactive *network* based simulation environments with graphical user interfaces (for example Corbin (1987) and Conway (1987)) have more flexibility in specifying model structures. Mice and pop-up menus are often used for this purpose. As with the template models, however, the accessibility of these environments comes with the disadvantage of limited flexibility. The benefit is a fairly short model building time, and some reasonable assurance that the model works as intended.

Users writing their simulations in a *simulation language* use model building blocks such as GENERATE, ADVANCE, TERMINATE, SEIZE, to specify the flow and logic of the model. One such model was shown in Figure 3. While simulation languages share many features with programming languages, they differ in that statements in a modeling language correspond to activities in the system of interest (WAIT, QUEUE) rather than to activities in the computer (multiply, divide).

Some situations are so unique that they cannot be effectively modeled using any of the approaches listed above. While these situations are quite rare, they occasionally occur when complex material handling systems using fairly elaborate control schemes are simulated. In this case general purpose *programming languages* must be used. Thesen (1987) gives an overview of how such programs can be written in Pascal. Approximately 1000 lines of Pascal code would be required to implement the simple model depicted in Figure 3.

The approaches discussed above are summarized in Table 5.

Model Type	Inter-face	User Input	Flexibility & Difficulty	Typical Tool
Template	Menus	Parameters	Very Low	StarCell Sandie
Network	Graph-ical	Structure Parameters	Low	XCell Simple_1
Simulation Language	Text editor	Structure Parameters Performance measures	High	GPSS, SIMAN, SIMSCRIPT SLAM
Programming Language	Text editor	Structure Parameters Performance measures Time keeping	Very High	C FORTRAN GASP Modula-2 PASCAL

Table 5: Four approaches to model representation.

Finally there are certain simulation systems that fall outside the classification scheme listed above. For example FACTROL installs simulation models that are integrated with a factory's production scheduling system. These systems aid in day-to-day scheduling by running simulations using current information from the shop floor. (MacFarland (1987)).

III. OUTPUT ANALYSIS

A. Analysis of a Single Design

Simulations are used to develop some sense of the performance of a system over time. We may want to know the average waiting time in a facility or we may want to know the expected demand during a stock out period. While this type of summary information is not explicitly used in the simulation model, it is usually easy to accumulate it during the run.

For example, we can easily record the length-of-stay for each individual customer during a run, and at the end of the run we can compute the *average* length of stay. Since random variates are used to replicate real life in service times and arrival times, the resulting average will also be a random variable. In other words simulation follows the RIRO principle -- random input, random output. It is essential to keep this in mind when interpreting your results. If we were to run the simulation again, we would in all likelihood observe a different average. *The purpose of most simulation runs is to estimate the true mean of such averages and to develop an understanding of their variability*.

B. Comparing Designs

Simulations are often used to compare the performance of different solutions to a problem. This requires careful planning. Since the observed value of a performance measure for any single simulation run may be thought of as a random variable, the observed difference between the performances of systems will also be a random variable. Careful analysis is therefore necessary to see if this performance difference is statistically significant. This analysis can answer questions such as the following:

One decision variable
- Which *system* has the fastest response time?
- Is throughput increased if the *buffer size* is increased 10%?
- What is the optimal *size* of my maintenance crew?

Two decision variables
- Does throughput change when scheduling *rules* and *buffer sizes* change?
- What is the best combination of *order-point* and *order-quantity* ?

Many decision variables
- What *factors* affect throughput?
- What *staffing pattern, truck routing* and *fire station territories* minimize response time?

An Example

A simulation study was conducted to compare two possible decision rules for determining the order in which jobs were processed in a simple three station job-shop. The *First Come First Served* (FCFS) and the *Shortest Processing Time First* (SPT) dispatching rule were compared. The two models were run 10 times each. The following results were obtained:

Replication	FCFS	SPT
1	113.9	115.2
2	121.7	112.8
3	117.5	114.1
4	121.4	110.8
5	114.5	117.5
6	118.6	114.3
7	118.8	114.9
8	113.2	113.9
9	113.9	118.1
10	118.1	113.9
Average	117.2	114.6
Standard Deviation	9.9	4.4

Table 6: Mean time in the system for 10 replications of two different scheduling policies for a three station flow-shop.

If we are justified in assuming identical variablility in the process producing the two data sets, then we can estimate the 90% confidence interval about the difference between the two true averages as

$$-2.00 \leq (\mu_1 - \mu_2) \leq 7.24 .$$

Since this interval includes zero, we are unable to reject the hypothesis that the two service policies result in identical performance.

This failure is either due to the fact that there indeed is no difference in performance, or, it is due to the fact that the variance in the observed data was too high. If the variability were due to the built-in randomness of the model, then we could use *variance reduction* techniques to reduce this variability.

One approach to reducing variability is to operate the two different models under identical random conditions. For example the models could be operated such that parts would arrive at identical points in time in the two models. Performance differences would then not be due to random differences in arrival times. We then make *pair-wise* comparisons between replications. The resulting differences are of course random variables, and their true means must be the same as the difference between the true means estimated in Table 6. However, since we have eliminated a source of variability, it is likely that their variance is smaller. As we show in Table 7 this indeed is the case for our example.

Replication	FCFS	SPT	diff
1	113.9	110.8	3.1
2	121.7	117.5	4.2
3	117.5	114.3	3.2
4	121.4	114.9	6.5
5	114.5	113.9	0.6
6	118.6	112.8	5.8
7	118.8	114.1	4.7
8	113.2	115.2	-2.0
9	113.9	118.1	-4.2
10	118.1	113.9	4.2
Average			2.61
Standard Deviation			1.86

Table 7: Differences between mean time in the system for 10 replications of two different scheduling policies for a three station flow-shop using identical arrival times for each pair of replications.

The corresponding confidence interval is

$$0.59 \leq (\mu_1 - \mu_2) \leq 4.68$$

Since this confidence interval does not include zero, we reject the hypothesis that the two policies result in identical performance, enabling us to recommend one policy over the other. Using this *common random numbers technique* we reduced the inherent randomness in the modeling process by employing common random number streams in the two scenarios. The resulting lower variance enabled us to gain more information from our simulation. A brief survey of other variance reduction techniques is given in Nelson(1987).

C. Another Pitfall

Most conventional statistical data analysis techniques require that the data be independent and identically distributed random variables. Simulation data almost never satisfies this assumption. For example, the best predictor of the waiting time for customer 23 in a busy system may be the waiting time experienced by customer 22. Intuitively, if adjacent values in a data set tend to be similar, the data is said to be positively autocorrelated. Autocorrelated data does not satisfy the assumption of independence. Hence blind application of conventional statistical techniques will lead to misleading results whenever the real-life application shows autocorrelation.

Since changes occur more slowly in positively autocorrelated data, the estimated sample variance is usually less than the true variance. Confidence intervals based on low variance estimators are too narrow, and they may lead us to believe that our simulation results include much less error than they actually do. This type of error could lead an analyst to believe that a given change in the system would cause a large improvement in performance when, actually, the improvement observed in the model was simply random variation. The resulting policy recommendation could be an expensive mistake.

Several methods have been developed to deal with autocorrelated data. Many practitioners use *batch means analysis* to develop confidence limits about the mean for autocorrelated data. The original data set is replaced with a reduced data set containing only the means of contiguous batches of the original observations. The variability in this set is then used to estimate the desired confidence interval. This technique is based on the hope that the means formed by these contiguous groups (or batches) of observations are uncorrelated. The autocorrelation in the data is not completely eliminated by this technique, but it can be reduced by increasing total sample size and changing the batch sizes . Hence the accuracy of a confidence interval increases as the length of the run increases. While beyond the scope of this tutorial, more sophisticated transformations are generally more efficient than the batch means technique, in that they yield more information from a data set of the same size. Hence shorter, less costly runs may suffice for the more sophisticated techniques.

In sum, since performance measures generated by simulations are random , it is often difficult to estimate their true means. Blind application of inappropriate statistical techniques may lead to misleading conclusions and hence expensive errors in policy.

IV. FINAL REMARKS

In this tutorial, we have attempted to provide an introduction to the uses of simulation, the underlying concepts, and the types of computer packages available to the analyst. While some of these tools require significant expertise and experience, others are quite accessible to the novice.

We have highlighted a few guidelines for the beginning analyst:

1) Define your objectives before simulating.
2) Use the correct level of detail -- begin with a simple model.
3) Select software that is appropriate for your problem, level of experience, and time frame.
4) Remember that simulation results are observations of random variables, and interpret your results accordingly.

We have also pointed to a few of the many technical considerations involved in effective simulation, but, needless to say, in this short tutorial, the list of subjects that we treated very lightly or omitted completely is very long. Discussions on many such subjects can be found elsewhere in this volume In addition, we identify sources of additional information in the bibliography section.

BIBLIOGRAPHY

General Texts

Banks, Jerry and John S. Carson II (1984) ***Discrete-Event System Simulation***, Prentice-Hall, Englewood Cliffs,N.J.,1984

Bradey, Paul, Bennett L. Fox, and Linus E. Schrage (1987) ***A Guide to Simulation, 2nd edition.*** Springer-Verlag, NY.

Bulgren, William G. (1982) ***Discrete System Simulation***, Prentice-Hall,Englewood Cliffs, NJ.

Fishman,G.S. (1973) ***Concepts and Methods in Discrete Event Digital Simulation***, Wiley, NY.

Fishman, G.S. (1978) ***Principles of Discrete Event Simulation***, Wiley, NY.

Gottfried Byron S. (1984) ***Elements of Stochastic Process Simulation***, Prentice-Hall, Englewood Cliffs, NJ.

Gordon, Geoffrey (1978) ***System Simulation***, 2nd ed.,Prentice-Hall, Englewood Cliffs, NJ.

Graybeal, W. J., and U. W. Pooch (1980) ***Simulation: Principles and Methods***, Winthrop, Cambridge, MA.

Johnson, M.E. (1987) ***Multivariate Statistical Simulation***, John Wiley, NY.

Law, Averill M., and W. David Kelton (1982) ***Simulation Modeling and Analysis***, McGraw-Hill, NY.

Nayor,T.H. (1971) ***Computer Simulation Experiments with Models of Economic Systems,*** Wiley, NY

Payne, James A. (1982) ***Introduction To Simulation: Programming Techniques and Methods of Analysis,*** McGraw-Hill, NY.

Shannon, Robert E. (1975)***System Simulation: The Art and Science,*** Prentice-Hall, Englewood Cliffs, NJ.

Environments and Languages

Cobbin, Philip (1988) **The SIMPLE_1 Simulation Environment,** *Proceedings of the 1987 Winter Simulation Conference, A. Thesen, H. Grant and W.David Kelton (eds.)* pp 136 -145.

Conway, R. and W. Maxwell (1987) **Modeling Ascyncronous Material Handling in XCELL+** , *Proceedings of the 1987 Winter Simulation Conference, A. Thesen, H. Grant and W.David Kelton (eds.)* pp 202 - 206.

Cox, S. (1986) ***GPSS/PC Users Manual*** MINUTEMAN SOFTWARE, Stowe, MA.

Gordon, Geoffrey (1975)***The Application of GPSS V to Discrete System Simulation,*** Prentice-Hall, Cliffs,NJ.

Henriksen, J.O and R.C.Crain (1983) ***GPSS/H Users Manual 2nd ed.*** Wolverine Software Corp, Annandale, VA.

MacFarland, Douglas G. (1987) ***Scheduling Manufacturing Systems with FACTOR,*** *Proceedings of the 1987 Winter Simulation Conference, A. Thesen, H. Grant and W.David Kelton (eds.) pp 235-237.*

Markowits, H.M. and P.J.Kiviat, and R. Villanueva (1987) ***Simscript II.5 Programming Language.*** CACI. Los Angeles, CA.

Nelson, B.L. (1987) ***Variance Reduction for Simulation Practitioners,*** *Proceedings of the 1987 Winter Simulation Conference, A. Thesen, H. Grant and W.David Kelton (eds.)* pp43-51.

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

Pritsker, A.A.B, C.Elliot Sigal and R.D.Jack Hammesfahr (1988)***SLAM II Network Models for Decision Support,*** Prentice Hall, NJ.

Pritsker, A.A.B (1986) ***Introduction to Simulation and SLAM III, 3rd edition,*** Systems Publishing Corp.

Standridge, C.R., and A.A.B. Pritsker (1987) ***TESS: The Extended Simulation System,*** Halsted Press.

Steudel, H.J. and T.Park (1987) ***STAR*CELL: A Flexible Manufacturing Cell Simulator,*** *Proceedings of the 1987 Winter Simulation Conference, A. Thesen, H. Grant and W.David Kelton (eds.)* pp230-234.

Suri,R. and M.Tomsicek (1988) ***Rapid Modeling Tools for Manufacturing Simulation and Analysis,*** *Proceedings of the 1988 Winter Simulation Conference.*

Thesen, Arne (1987) ***Writing Simulations from Scratch: Pascal Implementations,*** *Proceedings of the 1987 Winter Simulation Conference, A. Thesen, H. Grant and W.David Kelton (eds.)* pp152-164.

Analysis

Box, G.E.P, W.G.Hunter and J.S.Hunter (1978) ***Statistics for Experimenters,*** Wiley and Sons, NY.

Johnson, M.E. (1987) ***Multivariate Statistical Simulation,*** John Wiley, NY.

Meketon, Marc S. (1987) ***Optimization in Simulation: A Survey of Recent Results.*** *Proceedings of the 1987 Winter Simulation Conference, A. Thesen, H. Grant and W.David Kelton (eds.)* pp 58-67.

Nelson, B.L. (1987) ***Variance Reduction for Simulation Practitioners,*** *Proceedings of the 1987 Winter Simulation Conference, A. Thesen, H. Grant and W.David Kelton (eds.)* pp43-51.

Sargent, R. (1987) ***An Overview of Verification and Validation of Simulation Models,*** *Proceedings of the 1987 Winter Simulation Conference, A. Thesen, H. Grant and W.David Kelton (eds.)* pp 33-39.

Thesen, Arne (1985), ***Introduction to SANDIE,*** *Proceedings of the 1985 Winter Simulation Conference,.* D.Gantz, G. Blais and S.Solomon (eds) pp508-515.

AUTHORS BIOGRAPHY

ARNE THESEN, a Professor of Industrial Engineering and Computer Sciences at the University of Wisconsin-Madison, wrote his first simulation program in 1964. His current research interests are in the area of expert scheduling systems.

ARNE THESEN
Department of Industrial Engineering
1513 University Ave,
Madison, WI 53706
(608) 262-3960.

LAUREL E. TRAVIS is a recent Ph.D. in Operations Research and an Assistant Professor in the Management Department at the University of Wisconsin-Whitewater.

Laurel Travis
Management Department
University of Wisconsin-Whitewater
Whitewater, WI 53190