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

Applying the simulation process

Jerry Banks
School of Industrial and
Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

John S. Carson, II
Carson/Banks & Associates
4279 Roswell Road
Suite 604-173
Atlanta, Georgia 30342

ABSTRACT

The purpose of this tutorial is to explain the steps required in conducting an entire simulation project. We first explain the twelve steps we consider necessary to accomplish a thorough simulation study. Then we describe three projects that the authors have completed. We discuss the systems, model assumptions, and input data for these projects. During the tutorial, we illustrate how the twelve steps pertain to the three simulation studies.

1. INTRODUCTION

The purpose of this tutorial is to explain and illustrate the main steps in a simulation study. Three actual simulation projects will be used to illustrate the steps in a study. These projects involved studies of the following systems:

(1) Clinics at a major midwestern hospital;
(2) Cigarette manufacturing at one of the larger firms in the cigarette industry;
(3) Warehouse/sales centers for a major beverage distributor.

With these three studies, we will illustrate the 12 steps outlined in the text by Banks and Carson (1984). In addition, we will discuss the important issue of user interface to the model.

In the next section of this paper, we review the twelve steps in a simulation study. In the last section, we briefly describe the three systems, modeling assumptions, and input data. In the third section, we briefly illustrate the steps by at least one of the three simulation studies.

2. STEPS IN A SIMULATION STUDY

The twelve steps in a simulation study, as outlined by Banks and Carson (1984), are as follows:

1. **Problem Formulation.** Every study begins with a statement of the problem. It is important that both the analyst and the policymaker agree with the formulation. The problem statement may be reformulated as the solution progresses.

2. **Setting of Objectives and Overall Plan.** The objectives indicate the questions to be answered by the simulation. Is simulation the appropriate methodology? What are the individual tasks to be accomplished? How many resources are allocated to each task?

3. **Model Building.** This step is probably as much an art as a science. Start with a simple model and build toward greater complexity. Determine an appropriate border for the environment being simulated. Involve the model user in the model construction. Only experience with real systems can "teach" the art of model building.

4. **Data Collection.** Data collection takes a large amount of resources. The collection of data must begin early in the simulation study as the time required may be lengthy. The objectives of the study dictate the kind of data to be collected. Data needs may change as the model grows in complexity.

5. **Coding.** The model must be programmed for a computer. Determine the language to be used; general-purpose language such as FORTRAN; special-purpose simulation language such as SLAM®, SIMAN®, GPSS, or SIMSCRIPT®, or a simulator such as MAP/1™, SIMFACTORY®, or AutoMod™. Each category of language has its advantages and disadvantages.

6. **Verification.** Determine if the computer program is performing properly. If the input parameters and logical structure of the model are correctly represented in the code, verification has been completed.

7. **Validation.** Determine that the model is an accurate representation of the system. Calibration is required, i.e., compare the model to the real system and make appropriate changes. The process is repeated until model accuracy is judged acceptable.

8. **Experimental Design.** Determine the alternatives that are to be simulated. For each system design to be simulated, decisions need to be made concerning the initialization period, the length of the runs, and the number of replications to be made.

9. **Production Runs and Analysis.** Necessary to estimate measures of performance for the system designs that are being simulated.

10. **Decide on More Runs.** Based on the analysis of runs that have been completed, the analyst determines if additional runs are needed and what design these additional experiments should follow.

11. **Document Program and Report Results.** If the model is modified in the future, adequate documentation can greatly enhance the effort. Documentation also increases confidence so that model users and policymakers can make decisions with confidence based on the output.

12. **Implementation.** Contingent on how well the previous eleven steps have been accomplished. Very sensitive to the involvement of the ultimate user during the entire simulation process.

In addition to the 12 steps discussed above, another issue important for the model's success is the user interface. The direct user of the model may be the model developer or a client. How does the user interact with the model to obtain runs, to enter new input data, and to get desired reports? Can the user see the model dynamics? That is, is the model animated or does it at least display some output graphically as it is running?

First, we will consider user interaction on the input side. At one end of the spectrum, in a model written entirely in a language such as GPSS or the network portion of SLAM, the data and the model are contained in the same file. To change the input data, the user must have some rudimentary knowledge of GPSS or SLAM, and also must be very careful not to make accidental changes to the model file. In this regard, SIMAN lies a step up on the spectrum: it separates the data (or most of it) in a separate file called the experimental frame from the model itself which is in a file called the model frame. While some knowledge of SIMAN is necessary to change the data in the experimental frame, there is no chance of a user accidentally changing the model structure.

As a third step up on the spectrum, models written in GPSS, SLAM, SIMAN or most other languages can read input data from a separate input data file. (Depending on the language, there may be limitations on the parameters that can be assigned from input data.) With a separate data file, there is no chance of a user accidentally corrupting a model's structure.

The highest level of user interface would involve a combination of an interactive program to prompt for input, edit and verify its correctness and consistency (to the greatest degree possible), and if appropriate and possible, to read data from the client's existing databases.

We will illustrate both extremes of the user interface spectrum by our three studies.

3. THE SYSTEMS, THE MODEL ASSUMPTIONS, AND THE INPUT DATA

We give a brief description of the three systems, some of the key model assumptions, and some of the key input data:

3.1 Clinics at a Major Midwestern Hospital.

Several combinations of clinics were studied. In general, one of the systems studied contains from 1 to 4 clinics. Each clinic has its own set of senior staff physicians and residents. Each physician has some number of associated residents and some number of patients with scheduled appointments for a given day. The clinics share a bank of examination rooms. The model covers one day from 8:00 a.m. to 5:00 p.m., plus the additional time until all patients have been seen.

The clinics see from one to four types of patients. In addition, each clinic has no-shows and add-ons (i.e., non-scheduled patients).

Within each clinic, patients are scheduled according to a rule that takes into account the expected variation in exam duration as a function of patient type. In addition, physician's hours may vary from day to day, and physicians are supposed to get a lunch break.

Although not completely accurate, the typical patient flow is assumed to be as follows:

Patient arrives.
Patient queues until room is available.
Patient enters exam room.
Patient prepares for exam.
Patient waits until resident enters room.
Resident begins exam.
During first part of exam time, resident is alone with patient.
During last portion, resident confers with senior staff physician.
Resident and physician finish with patient.
Patient gets dressed and nurse cleans the room, after which room is available.
Meanwhile, resident performs other duties (summary dictation, etc.), after which resident is available for next patient.
Patient queueing discipline was fairly complex and is not discussed here. It depended upon the patient's actual arrival time versus scheduled appointment time, the patient's priority, how busy the patient's physician was, and other factors.

The input data needs for the model were extensive as indicated in the following paragraphs:

Actual arrival times of scheduled patients are generated in the model from the scheduled arrival times plus or minus early and late times based on input data specified as a statistical distribution.

Actual exam times are based on a statistical distribution of exam times, with a different distribution for each patient type and clinic.

For both early/late times and exam times, extensive time study data was available. This data was used to form empirical distributions. No attempt was made to fit the distribution to the data.

Some times, such as patient dressing time, were based on subjective estimates. In these cases, we used estimates of minimum, mode (most likely) and maximum times. These three parameters were used to define a triangular distribution.

For a given set of input data (number of physicians and residents, physician schedule, number of exam rooms, scheduling rule, etc.), the model predicts utilization of exam rooms (% of total time occupied or being cleaned, and number of patient exams), the number of patients seen by each physician and resident, the number of patients by clinic and patient type, and the queueing time of patients. Statistics are given for patient time in the reception room, total patient time in room, patient waiting time for a resident, patient waiting time for a physician, and total patient time with a resident.

3.2 Cigarette Manufacturing

The second study involved the cigarette fabrication process at a modern manufacturing facility at one of the larger cigarette manufacturers. The cigarette fabrication process consists of a maker/filter tip attachemnt (FTA), reservoir, packer, pack downdrop and pack conveyor, and wrapper/cartoner. What follows is a simplified description of the system. (For further reading, see Carson, Wysowski, Carroll, and Wilson [1981].)

The maker/FTA fabricates a continuous rod of tobacco, wraps paper around the rod, and cuts it into the desired length. The FTA applies a filter to the tobacco section. The cigarette is conveyed through a reservoir to a packer, which places foil and a label around groups of 20 cigarettes. Packs are then conveyed to the wrapper/cartoner.

Modern makers operate at speeds from about 4000 to over 7000 cigaretts per minute. Makers and packers are subject to fairly frequent product-induced failures (for example, the cigarette paper tears, or glue in the packer gets in the wrong place). These occur from one to perhaps fifteen times per hour, with downtimes lasting from a few seconds to a few minutes. The reservoir acts as a buffer between the maker and the packer. When the packer is down, the maker can continue to produce into the reservoir until it is full. Most packers have an overspeed capability which allows them to catch up. Similarly, when the maker goes down, the packer can continue to pack cigarettes from the reservoir. As stopping and starting the packer tends to reduce quality and yield, continuous operation of the packer is most desirable. Of course, to maximize production, the maker must be kept running as much as possible.

The objective of the modeling project was to determine the "optimum" capacity of the reservoir, given a particular configuration of machines, their processing rates and downtime characteristics. The performance measure of greatest interest was production yield over an appropriate period of time. Measures used in the validation and effort included production yield, number of times and duration of reservoir filling, and number of times and duration of reservoir emptying.

Discussions with plant engineers, maintenance personnel, mechanics and operators lead to development of a process flow diagram and a list of detailed assumptions on process operation. Meetings and actual observation of the equipment were used to resolve differences of opinion on process operation. A simulation model was developed in GPSS V.

With period updates of the most important data, this model has been used by B & W engineers to size the reservoir (and for other purposes) numerous times since 1980 whenever makers and packers with different processing speeds and downtime characteristics were being considered for purchase. The model-generated data is combined with accounting and cost data to aid in choosing the most economic combination of equipment. Time studies on fabrication groups with the model suggested "optimum" reservoir capacity have appeared to confirm model predictions, further increasing model credibility.

3.3 Warehouse/Sales Centers

The goal of this project was to develop a generic warehouse simulator that would model any of the client's facilities. For this reason, a number of different warehouses were observed.

The first phase (discussed here) concentrated on forklift activity and the storage and retrieval of full pallet loads of product. Typically, product arrives either by transport from an off-site bottling plant, or arrives by palletizer and conveyor from an on-site bottling plant. Either a single or double-pallet forklift stores the product. The same forklift must keep the depalletizer supplied with empty bottles or new cans. Forklifts also must supply a centralized picking area called the mixed pallet makeup area (MPMA). In addition, forklifts take full unit loads to staging areas and onto transport trucks, and load the route trucks with mixed pallets from the MPMA.

The user interface for input data is more sophisticated and user friendly than any of the other models. The user draws the layout of the warehouse using a CAD-like front end. All travel distances are computed automatically from this graphical input. In addition, the user is prompted for all other needed input. For example, with the layout on screen, the user can specify which products will be stored in each area of the warehouse. The user can build an extensive database of input files, define various scenarios by choosing combination of input files, define initial and run conditions, define any number of statistical measurement periods for output reports, and run the simulation. The model is completely data driven, but the input interface gives the user numerous choices, including any desired warehouse layout and placement of product in storage areas.

The user interface and model are both written in Pascal. The model is based on SIMTOOLS, a collection of routines written in Pascal to support discrete-event simulation (Seila, 1986). The user interface is based on MS Windows.

REFERENCES

Banks, J. and Carson, J. S., (1984). Discrete-Event System Simulation, Prentice-Hall, Englewood Cliffs, New Jersey.

Carson, J. S., Wilson, N., Carroll, D. and Wysowski, C. H., (1981). "A Discrete Simulation of a Cigarette Fabrication Process," Proceedings of the Twelfth Modeling and Simulation Conference, University of Pittsburgh.

Carson, J. S., et al, (1983). "The Use of Simulation Modeling at Brown & Williamson Tobacco Corporation," Proceedings of the Annual Industrial Engineering Conference.

Seila, A. F., (1986), "Discrete Event Simulation in Pascal with SIMTOOLS," Proceedings of the 1986 Winter Simulation Conference, J. Wilson, J. Henriksen, S. Roberts (eds.).

AUTHORS' BIOGRAPHIES

JERRY BANKS is Associate Professor of Industrial and Systems Engineering. His area of interest is simulation languages and modeling. He is the author of numerous articles and books. His latest book is Principles of Quality Control, published by John Wiley & Sons. Recent books include Discrete-Event Simulation, co-authored with John S. Carson, II, published by Prentice-Hall in 1984, Handbook of Tables and Graphs for the Industrial Engineer and Manager, co-authored with Russell G. Heikes, published by Reston Publishing Co., a division of Prentice-Hall also appearing in 1984, IBM PC Applications for the Industrial Engineer and Manager, co-authored with J. P. Spoerer and R. L. Collins, and published by Reston in 1986, and Procurement and Inventory Systems Analysis, co-authored with W. J. Fabrycky, published by Prentice-Hall in 1987. Dr. Banks received his Ph.D. in Engineering from the

Oklahoma State University in 1966. He is an active member of many technical organizations. He serves as IIE's representative to the Board of Winter Simulation Conference and as Vice Chairman of the Board.

Jerry Banks
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 39332, U.S.A.
(404) 894-2312

JOHN S. CARSON, II is a simulation consultant to a number of manufacturing and service industries. He has served on the faculties of the Georgia Institute of Technology and the University of Florida. He is co-author, with Jerry Banks, of the text Discrete-Event Systems Simulation. He is a member of IIE, ORSA and TIMS. Dr. Carson earned his BA in mathematics from the University of Virginia and his MS in mathematics and his Ph.D. (1978) in engineering from the University of Wisconsin at Madison.

John S. Carson, II
Carson/Banks & Associates
4279 Roswell Road
Suite 604-173
Atlanta, Georgia 30342, U.S.A.
(404) 971-1256