

Discrete and continuous systems were defined in Section 1.7. Discrete and continuous models are defined in an analogous manner. However, a discrete simulation model is not always used to model a discrete system, nor is a continuous simulation model always used to model a continuous system. Tanks and pipes might be modeled discretely, even though we know that fluid flow is continuous. In addition, simulation models may be mixed, both discrete and continuous. The choice of whether to use a discrete or continuous (or both discrete and continuous) simulation model is a function of the characteristics of the system and the objective of the study. Thus, a communication channel could be modeled discretely if the characteristics and movement of each message were deemed important. Conversely, if the flow of messages in aggregate over the channel were of importance, modeling the system via continuous simulation could be more appropriate. The models emphasized in this text are dynamic, stochastic, and discrete.

### 1.11 Discrete-Event System Simulation

This is a textbook about discrete-event system simulation. Discrete-event system simulation is the modeling of systems in which the state variable changes only at a discrete set of points in time. The simulation models are analyzed by numerical rather than analytical methods. *Analytical* methods employ the deductive reasoning of mathematics to “solve” the model. For example, differential calculus can be used to compute the minimum-cost policy for some inventory models. *Numerical* methods employ computational procedures to “solve” mathematical models. In the case of simulation models, which employ numerical methods, models are “run” rather than solved—that is, an artificial history of the system is generated from the model assumptions, and observations are collected to be analyzed and to estimate the true system performance measures. Real-world simulation models are rather large, and the amount of data stored and manipulated is vast, so such runs are usually conducted with the aid of a computer.

In summary, this textbook is about discrete-event system simulation in which the models of interest are analyzed numerically, usually with the aid of a computer.

### 1.12 Steps in a Simulation Study

Figure 1.3 shows a set of steps to guide a model builder in a thorough and sound simulation study. Similar figures and discussion of steps can be found in other sources [Shannon, 1975; Gordon, 1978; Law, 2007]. The number beside each symbol in Figure 1.3 refers to the more detailed discussion in the text. The steps in a simulation study are as follows:

- 1. Problem formulation** Every study should begin with a statement of the problem. If the statement is provided by the policymakers or those that have the problem, the analyst must ensure that the problem being described is clearly understood. If a problem statement is being developed by the analyst, it is important that the policymakers understand and agree with the formulation. Although not shown in Figure 1.3, there are occasions where the problem must be reformulated as the study progresses. In many instances, policymakers and analysts are aware that there is a problem long before the nature of the problem is known.

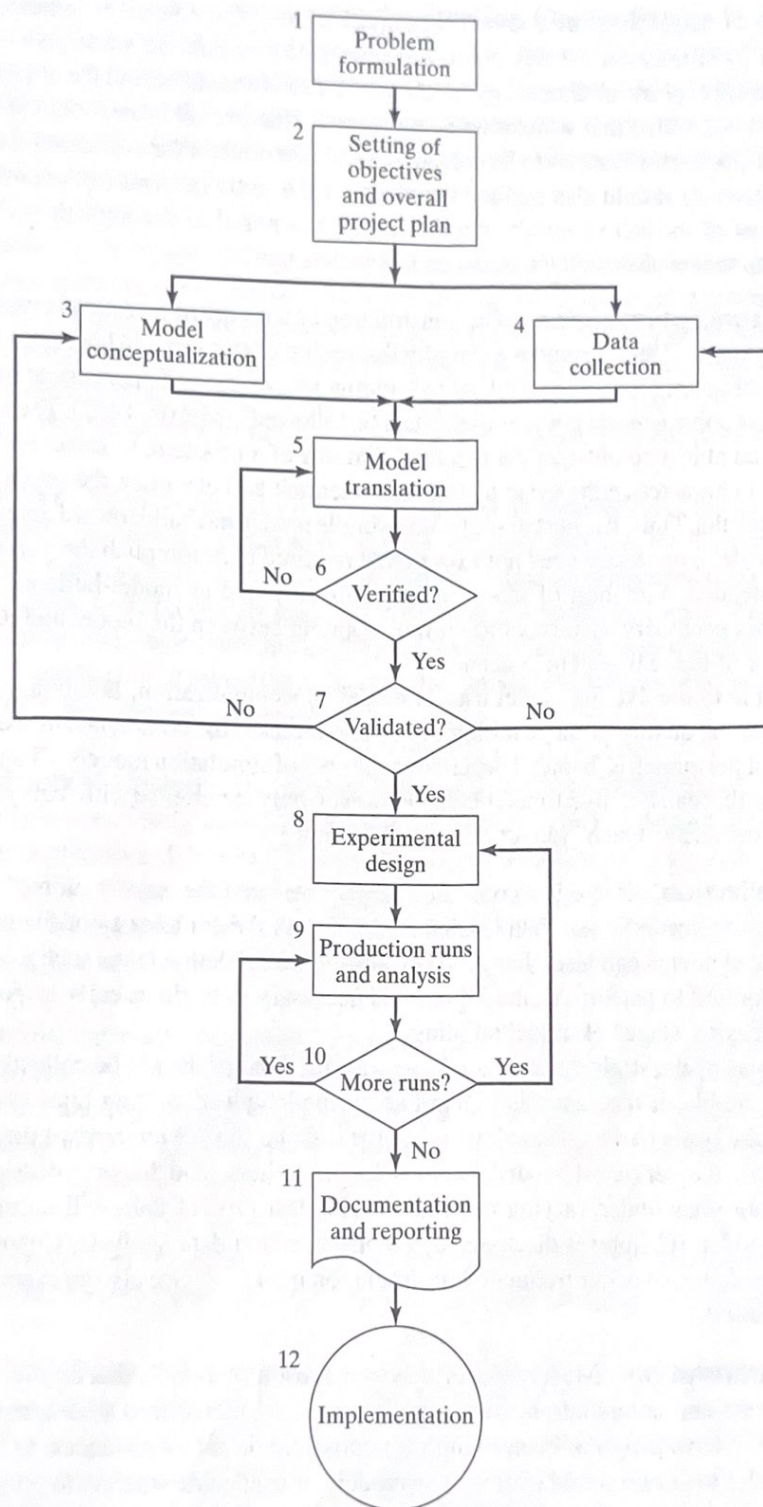


Figure 1.3 Steps in a simulation study.



**2. Setting of objectives and overall project plan** The objectives indicate the questions to be answered by simulation. At this point, a determination should be made concerning whether simulation is the appropriate methodology for the problem as formulated and the objectives as stated. Assuming that it is decided that simulation is appropriate, the overall project plan should include a statement of the alternative systems to be considered and of a method for evaluating the effectiveness of these alternatives. It should also include the plans for the study in terms of the number of people involved, the cost of the study, and the number of days required to accomplish each phase of the work, along with the results expected at the end of each stage.

**3. Model conceptualization** The construction of a model of a system is probably as much art as science. Pritsker [1998] provides a lengthy discussion of this step. "Although it is not possible to provide a set of instructions that will lead to building successful and appropriate models in every instance, there are some general guidelines that can be followed" [Morris, 1967]. The art of modeling is enhanced by an ability to abstract the essential features of a problem, to select and modify basic assumptions that characterize the system, and then to enrich and elaborate the model until a useful approximation results. Thus, it is best to start with a simple model and build toward greater complexity. However, the model complexity need not exceed that required to accomplish the purposes for which the model is intended. Violation of this principle will only add to model-building and computer expenses. It is not necessary to have a one-to-one mapping between the model and the real system. Only the essence of the real system is needed.

It is advisable to involve the model user in model conceptualization. Involving the model user will both enhance the quality of the resulting model and increase the confidence of the model user in the application of the model. (Chapter 2 describes a number of simulation models. Chapter 6 describes queueing models that can be solved analytically. However, only experience with real systems—versus textbook problems—can "teach" the art of model building.)

**4. Data collection** There is a constant interplay between the construction of the model and the collection of the needed input data [Shannon, 1975]. As the complexity of the model changes, the required data elements can also change. Also, since data collection takes such a large portion of the total time required to perform a simulation, it is necessary to begin as early as possible, usually together with the early stages of model building.

The objectives of the study dictate, in a large way, the kind of data to be collected. In the study of a bank, for example, if the desire is to learn about the length of waiting lines as the number of tellers changes, the types of data needed would be the distributions of interarrival times (at different times of the day), the service-time distributions for the tellers, and historic distributions on the lengths of waiting lines under varying conditions. This last type of data will be used to validate the simulation model. (Chapter 9 discusses data collection and data analysis; Chapter 5 discusses statistical distributions that occur frequently in simulation modeling. See also an excellent discussion by Henderson [2003].)

**5. Model translation** Most real-world systems result in models that require a great deal of information storage and computation, so the model must be entered into a computer-recognizable format. We use the term *program* even though it is possible, in many instances, to accomplish the desired result with little or no actual coding. The modeler must decide whether to program the model in a simulation language, such as GPSS/H<sup>TM</sup> (discussed in Chapter 4), or to use special-purpose



simulation software. For manufacturing and material handling, Chapter 4 discusses such software as AnyLogic®, Arena®, AutoMod™, Enterprise Dynamics®, Extend™, Flexsim, ProModel®, and SIMUL8®. Simulation languages are powerful and flexible. However, if the problem is amenable to solution with the simulation software, the model development time is greatly reduced. Furthermore, most simulation software packages have added features that enhance their flexibility, although the amount of flexibility varies greatly.

**6. Verified?** Verification pertains to the computer program that has been prepared for the simulation model. Is the computer program performing properly? With complex models, it is difficult, if not impossible, to translate a model successfully in its entirety without a good deal of debugging; if the input parameters and logical structure of the model are correctly represented in the computer, verification has been completed. For the most part, common sense is used in completing this step. (Chapter 10 discusses verification of simulation models, and Sargent [2007] also discusses this topic.)

**7. Validated?** Validation usually is achieved through the calibration of the model, an iterative process of comparing the model against actual system behavior and using the discrepancies between the two, and the insights gained, to improve the model. This process is repeated until model accuracy is judged acceptable. In the previously mentioned example of a bank, data was collected concerning the length of waiting lines under current conditions. Does the simulation model replicate this system measure? This is one means of validation. (Chapter 10 discusses the validation of simulation models, and Sargent [2007] also discusses this topic.)

**8. Experimental design** The alternatives that are to be simulated must be determined. Often, the decision concerning which alternatives to simulate will be a function of runs that have been completed and analyzed. For each system design that is simulated, decisions need to be made concerning the length of the initialization period, the length of simulation runs, and the number of replications to be made of each run. (Chapters 11 and 12 discuss issues associated with the experimental design, and Sanchez [2007] discusses this topic extensively.)

**9. Production runs and analysis** Production runs and their subsequent analysis, are used to estimate measures of performance for the system designs that are being simulated. (Chapters 11 and 12 discuss the analysis of simulation experiments, and Chapter 4 discusses software to aid in this step, including AutoStat (in AutoMod), OptQuest (in several pieces of simulation software), and SimRunner (in ProModel).)

**10. More runs?** Given the analysis of runs that have been completed, the analyst determines whether additional runs are needed and what design those additional experiments should follow.

**11. Documentation and reporting** There are two types of documentation: program and progress. Program documentation is necessary for numerous reasons. If the program is going to be used again by the same or different analysts, it could be necessary to understand how the program operates. This will create confidence in the program, so that model users and policymakers can make decisions based on the analysis. Also, if the program is to be modified by the same or a



different analyst, this step can be greatly facilitated by adequate documentation. One experience with an inadequately documented program is usually enough to convince an analyst of the necessity of this important step. Another reason for documenting a program is so that model users can change parameters at will in an effort to learn the relationships between input parameters and output measures of performance or to discover the input parameters that "optimize" some output measure of performance.

Musselman [1998] discusses progress reports that provide the important, written history of a simulation project. Project reports give a chronology of work done and decisions made. This can prove to be of great value in keeping the project on course. Musselman suggests frequent reports (monthly, at least) so that even those not involved in the day-to-day operation can be kept abreast. The awareness of these others can often enhance the successful completion of the project by surfacing misunderstandings early, when the problem can be solved easily. Musselman also suggests maintaining a project log to provide a comprehensive record of accomplishments, change requests, key decisions, and other items of importance.

On the reporting side, Musselman suggests frequent deliverables. These may or may not be the results of major accomplishments. His maxim is that "it is better to work with many intermediate milestones than with one absolute deadline." Possibilities prior to the final report include a model specification, prototype demonstrations, animations, training results, intermediate analyses, program documentation, progress reports, and presentations. He suggests that these deliverables should be timed judiciously over the life of the project.

The results of all the analysis should be reported clearly and concisely in a final report. This will allow the model users (now the decision makers) to review the final formulation, the alternative systems that were addressed, the criteria by which the alternatives were compared, the results of the experiments, and the recommended solution(s) to the problem. Furthermore, if decisions have to be justified at a higher level, the final report should provide a vehicle of certification for the model user/decision maker and add to the credibility of the model and of the model-building process.

**12. Implementation** The success of the implementation phase depends on how well the previous eleven steps have been performed. It is also contingent upon how thoroughly the analyst has involved the ultimate model user during the entire simulation process. If the model user has been involved during the entire model-building process and if the model user understands the nature of the model and its outputs, the likelihood of a vigorous implementation is enhanced [Pritsker, 1995]. Conversely, if the model and its underlying assumptions have not been properly communicated, implementation will probably suffer, regardless of the simulation model's validity.

The simulation-model building process shown in Figure 1.3 can be broken down into four phases. The first phase, consisting of steps 1 (Problem formulation) and 2 (Setting of objective and overall design), is a period of discovery or orientation. The initial statement of the problem is usually quite "fuzzy," the initial objectives will usually have to be reset, and the original project plan will usually have to be fine-tuned. These recalibrations and clarifications could occur in this phase or perhaps will occur after or during another phase (i.e., the analyst might have to restart the process).

The second phase is related to model building and data collection and includes steps 3 (Model conceptualization), 4 (Data collection), 5 (Model translation), 6 (Verification), and 7 (Validation). A continuing interplay is required among the steps. Exclusion of the model user during this phase can have dire implications at the time of implementation.



The third phase concerns the running of the model. It involves steps 8 (Experimental design), 9 (Production runs and analysis), and 10 (More runs). This phase must have a comprehensively conceived plan for experimenting with the simulation model. A discrete-event stochastic simulation is, in fact, a statistical experiment. The output variables are estimates that contain random error, and therefore a proper statistical analysis is required. Such a philosophy is in contrast to that of the analyst who makes a single run and draws an inference from that single data point.

The fourth phase, implementation, involves steps 11 (Documentation and reporting) and 12 (Implementation). Successful implementation depends on continual involvement of the model user and on the successful completion of every step in the process. Perhaps the most crucial point in the entire process is Step 7 (Validation), because an invalid model is going to lead to erroneous results, which, if implemented, could be dangerous, costly, or both.

## REFERENCES

- BANKS, J. [2008], "Some Burning Questions about Simulation." *ICS Newsletter*, INFORMS Computing Society, Spring.
- BANKS, J., and R. R. GIBSON [1997], "Don't Simulate When: 10 Rules for Determining when Simulation Is Not Appropriate," *IIE Solutions*, September.
- GORDON, G. [1978], *System Simulation*, 2d ed., Prentice-Hall, Englewood Cliffs, NJ.
- HENDERSON, S. G. [2003], "Input Model Uncertainty: Why Do We Care and What Should We Do About It?" in *Proceedings of the Winter Simulation Conference*, eds. S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, New Orleans, LA, Dec. 7–10, pp. 90–100.
- KLEIJNEN, J. P. C. [1998], "Experimental Design for Sensitivity Analysis, Optimization, and Validation of Simulation Models," in *Handbook of Simulation*, ed. J. Banks, John Wiley, New York.
- LAW, A. M. [2007], *Simulation Modeling and Analysis*, 4th ed., McGraw-Hill, New York.
- MORRIS, W. T. [1967], "On the Art of Modeling," *Management Science*, Vol. 13, No. 12.
- MUSSELMAN, K. J. [1998], "Guidelines for Success," in *Handbook of Simulation*, ed. J. Banks, John Wiley, New York.
- NAYLOR, T. H., J. L. BALINTFY, D. S. BURDICK, and K. CHU [1966], *Computer Simulation Techniques*, Wiley, New York.
- PEGDEN, C. D., R. E. SHANNON, and R. P. SADOWSKI [1995], *Introduction to Simulation Using SIMAN*, 2d ed., McGraw-Hill, New York.
- PRITSKER, A. A. B. [1995], *Introduction to Simulation and SLAM II*, 4th ed., Wiley & Sons, New York.
- PRITSKER, A. A. B. [1998], "Principles of Simulation Modeling," in *Handbook of Simulation*, ed. J. Banks, John Wiley, New York.
- SANCHEZ, S.R. [2007], "Work Smarter, Not Harder: Guidelines for Designing Simulation Experiments," in *Proceedings of the 2007 Winter Simulation Conference*, eds. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, Washington, DC, Dec. 9–12, pp. 84–94.
- SARGENT, R.G. [2007], "Verification and Validation of Simulation Models," in *Proceedings of the 2007 Winter Simulation Conference*, eds. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, Washington, DC, Dec. 9–12, pp. 124–137.
- SHANNON, R. E. [1975], *Systems Simulation: The Art and Science*, Prentice-Hall, Englewood Cliffs, NJ.
- SHANNON, R. E. [1998], "Introduction to the Art and Science of Simulation," in *Proceedings of the Winter Simulation Conference*, eds. D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, Washington, DC, Dec. 13–16, pp. 7–14.