CHAPTER 1

Principles of Simulation

JERRY BANKS
Georgia Institute of Technology

1.1 INTRODUCTION

The purpose of this handbook is to provide a reference to important topics that pertain to discrete-event simulation. All the contributors to this volume, who are a mix from academia, industry, and software developers, are highly qualified. The book is intended for those who want to apply simulation to important problems. If you are new to simulation, reading this chapter will provide an overview to the remainder of the book. If you studied simulation several years ago, reading this chapter will provide a useful review and update. [Much of this introductory chapter is from Banks et al. (1995).]

Chapter 1 is essentially in three parts. The first part begins with a definition and an example of simulation. Then modeling concepts introduced in the example are presented. Four modeling structures for simulation are then presented. The second part of the chapter concerns subjective topics. First, the advantages and disadvantages of simulation are discussed. Then some of the areas of application are mentioned. Last, the steps in the simulation process are described. The third part of the chapter has four sections. Each of these sections introduces operational aspects of discrete-event simulation. The chapter concludes with a summary.

1.2 DEFINITION OF SIMULATION

Simulation is the imitation of the operation of a real-world process or system over time. Simulation involves the generation of an artificial history of the system and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system that is represented. Simulation is an indispensable problem-solving methodology for the solution of many real-world problems. Simulation is used to describe and analyze the behavior of a system, ask what-if questions about the real

Handbook of Simulation, Edited by Jerry Banks.
ISBN 0-471-13403-1

system, and aid in the design of real systems. Both existing and conceptual systems can be modeled with simulation.

***Example 1** (Ad Hoc Simulation)* Consider the operation of a one-teller bank where customers arrive for service between 1 and 10 minutes apart in time, integer values only, each value equally likely. The customers are served in between 1 and 6 minutes, also integer valued, and equally likely. Restricting the times to integer values is an abstraction of reality since time is continuous, but this aids in presenting the example. The objective is to simulate the bank operation, by hand, until 20 customers are served, and to compute measures of performance such as the percentage of idle time, the average waiting time per customer, and so on. Admittedly, 20 customers are far too few to draw conclusions about the operation of the system for the long run, but by following this example, the stage is set for further discussion in this chapter and subsequent discussion about using the computer for performing simulation.

To simulate the process, random interarrival and service times need to be generated. Assume that the interarrival times are generated using a spinner that has possibilities for the values 1 through 10. Further assume that the service times are generated using a die that has possibilities for the values 1 through 6.

Table 1.1 is called an *ad hoc simulation table*. The setup of the simulation table is for

Table 1.1 Ad Hoc Simulation

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Customer	Time Between Arrivals	Arrival Time	Service Time	Service Begins	Time Service Ends	Time in System	Idle Time	Time in Queue
1	—	0	2	0	2	2	0	0
2	5	5	2	5	7	2	3	0
3	1	6	6	7	13	7	0	1
4	10	16	5	16	21	5	3	0
5	6	22	6	22	28	6	1	0
6	2	24	4	28	32	8	0	4
7	9	33	3	33	36	3	1	0
8	1	34	4	36	40	6	0	2
9	10	44	1	44	45	1	4	0
10	3	47	3	47	50	3	2	0
11	5	52	1	52	53	1	2	0
12	2	54	2	54	56	2	1	0
13	3	57	3	57	60	3	1	0
14	5	62	6	62	68	6	2	0
15	4	66	2	68	70	4	0	2
16	3	69	6	70	76	7	0	1
17	7	76	4	76	80	4	0	0
18	8	84	5	84	89	5	4	0
19	7	91	3	91	94	3	2	0
20	7	98	1	98	99	1	4	0
						79	30	10

the purpose of this problem but does not pertain to all problems. Column (1), Customer, lists the 20 customers who arrive at the system. It is assumed that customer 1 arrives at time zero; thus a dash is indicated in row 1 of column (2), Time Between Arrivals. Rows 2 through 20 of column (2) were generated using the spinner. Column (3), Arrival Time, shows the simulated arrival times. Since customer 1 is assumed to arrive at time 0 and there is a 5-minute interarrival time, customer 2 arrives at time 5. There is a 1-minute interarrival time for customer 3; thus the arrival occurs at time 6. This process of adding the interarrival time to the previous arrival time is called *bootstrapping*. By continuing this process, the arrival times of all 20 customers are determined. Column (4), Service Time, contains the simulated service times for all 20 customers. These were generated by rolling the die.

Now simulation of the service process begins. At time 0, customer 1 arrived and immediately began service. The service time was 2 minutes, so the service period ended at time 2. The total time in the system for customer 1 was 2 minutes. The bank teller was not idle since simulation began with the arrival of a customer. The customer did not have to wait for the teller.

At time 5, customer 2 arrived and began service immediately, as shown in column (6). The service time was 2 minutes, so the service period ended at time 7, as shown in column (6). The bank teller was idle from time 2 until time 5, so 3 minutes of idle time occurred. Customer 2 spent no time in the queue.

Customer 3 arrived at time 6, but service could not begin until time 7, as customer 2 was being served until time 7. The service time was 6 minutes, so service was completed at time 13. Customer 3 was in the system from time 6 until time 13, or for 7 minutes, as indicated in column (7), Time in System. Although there was no idle time, customer 3 had to wait in the queue for 1 minute for service to begin.

This process continues for all 20 customers, and the totals shown in columns (7), (8) (Idle Time), and (9) (Time in Queue) are entered. Some performance measures can now be calculated as follows:

Average time in system = 79/20 = 3.95 minutes.

Percent idle time = (30/99)(100) = 30%.

Average waiting time per customer = 10/20 = 0.5 minute.

Fraction having to wait = 5/20 = 0.25.

Average waiting time of those who waited = 10/3 = 3.33 minutes.

This very limited simulation indicates that the system is functioning well. Only 25% of the customers had to wait. About 30% of the time the teller is idle. Whether a slower teller should replace the current teller depends on the cost of having to wait versus any savings from having a slower server.

This small simulation can be accomplished by hand, but there is a limit to the complexity of problems that can be solved in this manner. Also, the number of customers that must be simulated could be much larger than 20 and the number of times that the simulation must be run for statistical purposes could be large. Hence, using the computer to solve real simulation problems is almost always appropriate.

Example 1 raises some issues that are addressed in this chapter and explored more fully in the balance of the book. The issues include the following:

1. How is the form of the input data determined?
2. How are random variates generated if they follow statistical distributions other than the discrete uniform?
3. How does the user know that the simulation imitates reality?
4. What other kinds of problems can be solved by simulation?
5. How long does the simulation need to run?
6. How many different simulation runs should be conducted?
7. What statistical techniques should be used to analyze the output?

Each of these questions raises a host of issues about which many textbooks and thousands of technical papers have been written. Although an introductory chapter cannot treat all of these questions in the greatest detail, enough can be said to give the reader some insight that will be useful in understanding the framework of the remainder of the book.

1.3 MODELING CONCEPTS

There are several concepts underlying simulation. These include system and model, system state variables, entities and attributes, list processing, activities and delays, and the definition of discrete-event simulation. Additional information on these topics is available from Banks et al. (1996) and Law and Kelton (1991). The discussion in this section follows that of Carson (1993). Chapter 2 provides an extensive discussion of the topic.

1.3.1 System, Model, and Events

A *model* is a representation of an actual system. Immediately, there is a concern about the limits or boundaries of the model that supposedly represent the system. The model should be complex enough to answer the questions raised, but not too complex. Consider an *event* as an occurrence that changes the state of the system. In Example 1, events include the arrival of a customer for service at a bank and the completion of a service. There are both internal and external events, also called *endogenous events* and *exogenous events*, respectively. For example, an endogenous event in Example 1 is the beginning of service of the customer since that is within the system being simulated. An exogenous event is the arrival of a customer for service since that occurrence is outside the simulation. However, the arrival of a customer for service impinges on the system and must be taken into consideration.

In this book we consider *discrete-event simulation models*. (Chapter 2 describes continuous and combined discrete-continuous models.) These are contrasted with other types of models, such as mathematical models, descriptive models, statistical models, and input–output models. A discrete-event model attempts to represent the components of a system and their interactions to such an extent that the objectives of the study are met. Most mathematical, statistical, and input–output models represent a system's inputs and outputs explicitly but represent the internals of the model with mathematical or statistical relationships. An example is the mathematical model from physics,

$$\text{force} = \text{mass} \times \text{acceleration}$$

based on theory. Discrete-event simulation models include a detailed representation of the actual internals.

Discrete-event models are *dynamic*; that is, the passage of time plays a crucial role. Most mathematical and statistical models are *static*, in that they represent a system at a fixed point in time. Consider the annual budget of a firm. The budget resides in a spreadsheet. Changes can be made in the budget and the spreadsheet can be recalculated, but the passage of time is usually not a critical issue. Further comments will be made about discrete-event models after several additional concepts are presented.

1.3.2 System State Variables

The *system state variables* are the collection of all information needed to define what is happening within a system to a sufficient level (i.e., to attain the desired output) at a given point in time. The determination of system state variables is a function of the purposes of the investigation, so what may be the system state variables in one case may not be the same in another case, even though the physical system is the same. Determining the system state variables is as much an art as a science. However, during the modeling process, any omissions will readily come to light. (On the other hand, unnecessary state variables may be eliminated.)

Having defined system state variables, a contrast can be made between discrete-event models and continuous models based on the variables needed to track the system state. The system state variables in a discrete-event model remain constant over intervals of time and change value only at certain well-defined points called *event times*. *Continuous models* have system state variables defined by differential or difference equations, giving rise to variables that may change continuously over time.

Some models are mixed discrete-event and continuous. There are also continuous models that are treated as discrete-event models after some reinterpretation of system state variables, and vice versa. The modeling of continuous systems is not treated in this book.

1.3.3 Entities and Attributes

An *entity* represents an object that requires explicit definition. An entity can be dynamic in that it "moves" through the system, or it can be static in that it serves other entities. In Example 1 the customer is a dynamic entity, whereas the bank teller is a static entity.

An entity may have *attributes* that pertain to that entity alone. Thus attributes should be considered as local values. In Example 1, an attribute of the entity could be the time of arrival. Attributes of interest in one investigation may not be of interest in another investigation. Thus, if red parts and blue parts are being manufactured, the color could be an attribute. However, if the time in the system for all parts is of concern, the attribute of color may not be of importance. From this example it can be seen that many entities can have the same attribute or attributes (i.e., more than one part may have the attribute "red").

1.3.4 Resources

A *resource* is an entity that provides service to dynamic entities. The resource can serve one or more than one dynamic entity at the same time (i.e., operate as a parallel server). A dynamic entity can request one or more units of a resource. If denied, the requesting entity joins a queue or takes some other action (i.e., is diverted to another resource, is ejected from the system). (Other terms for queues are *files*, *chains*, *buffers*, and *waiting lines*.) If permitted to capture the resource, the entity remains for a time, then releases the resource. There are many possible states of a resource. Minimally, these states are idle and busy. But other possibilities exist, including failed, blocked, or starved.

1.3.5 List Processing

Entities are managed by allocating them to resources that provide service; by attaching them to event notices, thereby suspending their activity into the future; or by placing them into an ordered list. Lists are used to represent queues.

Lists are often processed according to FIFO (first in, first out), but there are many other possibilities. For example, the list could be processed by LIFO (last in, first out), according to the value of an attribute, or randomly, to mention a few. An example where the value of an attribute may be important is in SPT (shortest process time) scheduling. In this case the processing time may be stored as an attribute of each entity. The entities are ordered according to the value of that attribute, with the lowest value at the head or front of the queue.

1.3.6 Activities and Delays

An *activity* is a period of time whose duration is known prior to commencement of the activity. Thus, when the duration begins, its end can be scheduled. The duration can be a constant, a random value from a statistical distribution, the result of an equation, input from a file, or computed based on the event state. For example, a service time may be a constant 10 minutes for each entity; it may be a random value from an exponential distribution with a mean of 10 minutes; it could be 0.9 times a constant value from clock time 0 to clock time 4 hours, and 1.1 times the standard value after clock time 4 hours; or it could be 10 minutes when the preceding queue contains at most four entities and 8 minutes when there are five or more in the preceding queue.

A *delay* is an indefinite duration that is caused by some combination of system conditions. When an entity joins a queue for a resource, the time that it will remain in the queue may be unknown initially since that time may depend on other events that may occur. An example of another event would be the arrival of a rush order that preempts the resource. When the preempt occurs, the entity using the resource relinquishes its control instantaneously. Another example is a failure necessitating repair of the resource.

Discrete-event simulations contain activities that cause time to advance. Most discrete-event simulations also contain delays as entities wait. The beginning and ending of an activity or delay are events.

1.3.7 Discrete-Event Simulation Model

Sufficient modeling concepts have been defined so that a discrete-event simulation model can be defined as one in which the state variables change only at those discrete points in time at which events occur. Events occur as a consequence of activity times and delays. Entities may compete for system resources, possibly joining queues while waiting for an available resource. Activity and delay times may "hold" entities for durations of time.

A discrete-event simulation model is conducted over time ("run") by a mechanism that moves simulated time forward. The system state is updated at each event, along with capturing and freeing of resources that may occur at that time.

1.4 MODELING STRUCTURES

There are four modeling structures taken by the simulation community. They are known as the process-interaction method, event-scheduling method, activity scanning, and the three-phase method. The descriptions are rather concise; readers requiring greater explanation are referred to Balci (1988) or Pidd (1992). The first two of these modeling structure topics are discussed in Chapter 2. We describe all four of them briefly here.

1.4.1 Process-Interaction Method

The simulation structure that has the greatest intuitive appeal is the process-interaction method. The notion is that the computer program should emulate the flow of an object through the system. The entity moves as far as possible in the system until it is delayed, enters an activity, or exits from the system. When the entity's movement is halted, the clock advances to the time of the next movement of any entity.

This flow, or movement, describes in sequence all the states that the object can attain in the system. For example, in a model of a self-service laundry a customer may enter the system, wait for a washing machine to become available, wash his or her clothes in the washing machine, wait for a basket to become available to unload the washing machine, transport the clothes in the basket to a drier, wait for a drier to become available, unload the clothes into a drier, dry the clothes, and leave the laundry.

1.4.2 Event-Scheduling Method

The basic concept of the event-scheduling method is to advance time to when something next happens. This usually releases a resource (i.e., a scarce entity such as a machine or transporter). The event then reallocates available objects or entities by scheduling activities where they can now participate. For example, in the self-service laundry, if a customer's washing is finished and there is a basket available, the basket could be allocated immediately to the customer and unloading of the washing machine could begin. Time is advanced to the next scheduled event (usually the end of an activity) and activities are examined to see if any can now start as a consequence.

1.4.3 Activity Scanning

The third simulation modeling structure is activity scanning. It is also known as the two-phase approach. Activity scanning is similar to rule-based programming. (If a specified condition is met, a rule is *fired*, meaning that an action is taken.) Activity scanning produces a simulation program composed of independent modules waiting to be executed. Scanning takes place at fixed time increments at which a determination is made concerning whether or not an event occurs at that time. If an event occurs, the system state is updated.

1.4.4 Three-Phase Method

The fourth simulation modeling structure is known as the three-phase method. Time is advanced until there is a state change in the system or until something next happens. The system is examined to determine all of the events that take place at this time (i.e., all the activity completions that occur). Only when all resources that are due to be released at this time have been released is reallocation of these resources into new activities started in the third phase of the simulation. In summary, the first phase is time advance. The second phase is the release of those resources scheduled to end their activities at this time. The third phase is to start activities given the global picture about resource availability.

Possible modeling inaccuracies may occur with the last two methods, as discrete time slices must be specified. With computing power growing so rapidly, high-precision simulation will be utilized increasingly, and the error due to discretizing time may become an important consideration.

1.5 ADVANTAGES AND DISADVANTAGES OF SIMULATION*

Competition in the computer industry has led to technological breakthroughs that are allowing hardware companies to produce better products continually. It seems that every week another company announces its latest release, each with more options, memory, graphics capability, and power.

What is unique about new developments in the computer industry is that they often act as a springboard for related industries to follow. One industry in particular is the simulation-software industry. As computer hardware becomes more powerful, more accurate, faster, and easier to use, simulation software does, too.

The number of businesses using simulation is increasing rapidly. Many managers are realizing the benefits of utilizing simulation for more than just the one-time remodeling of a facility. Rather, due to advances in software, managers are incorporating simulation in their daily operations on an increasingly regular basis.

For most companies, the benefits of using simulation go beyond simply providing a look into the future. These benefits are mentioned by many authors (Banks et al., 1996; Law and Kelton, 1991; Pegden et al., 1995; Schriber, 1991) and are included in the following.

1.5.1 Advantages

1. *Choose correctly.* Simulation lets you test every aspect of a proposed change or addition without committing resources to their acquisition. This is critical, because once the hard decisions have been made, the bricks have been laid, or the material handling systems have been installed, changes and corrections can be extremely expensive. Simulation allows you to test your designs without committing resources to acquisition.

2. *Compress and expand time.* By compressing or expanding time, simulation allows you to speed up or slow down phenomena so that you can investigate them thoroughly. You can examine an entire shift in a matter of minutes if you desire, or you can spend 2 hours examining all the events that occurred during 1 minute of simulated activity.

*Reprinted with the permission of the Institute of Industrial Engineers, 25 Technology Park, Norcross, GA 30092, 770-449-0161. Copyright ©1998.

3. *Understand why.* Managers often want to know why certain phenomena occur in a real system. With simulation, you determine the answer to the "why" questions by reconstructing the scene and taking a microscopic examination of the system to determine why the phenomenon occurs. You cannot accomplish this with a real system because you cannot see or control it in its entirety.

4. *Explore possibilities.* One of the greatest advantages of using simulation software is that once you have developed a valid simulation model, you can explore new policies, operating procedures, or methods without the expense and disruption of experimenting with the real system. Modifications are incorporated in the model, and you observe the effects of those changes on the computer rather than on the real system.

5. *Diagnose problems.* The modern factory floor or service organization is very complex, so complex that it is impossible to consider all the interactions taking place in a given moment. Simulation allows you to better understand the interactions among the variables that make up such complex systems. Diagnosing problems and gaining insight into the importance of these variables increases your understanding of their important effects on the performance of the overall system.

The last three claims can be made for virtually all modeling activities, queueing, linear programming, and so on. However, with simulation the models can become very complex and thus have a higher fidelity [i.e., they are valid representations of reality (as discussed in Chapter 10)].

6. *Identify constraints.* Production bottlenecks give manufacturers headaches. It is easy to forget that bottlenecks are an effect rather than a cause. However, by using simulation to perform bottleneck analysis, you can discover the cause of the delays in work in process, information, materials, or other processes.

7. *Develop understanding.* Many people operate with the philosophy that talking loudly, using computerized layouts, and writing complex reports convinces others that a manufacturing or service system design is valid. In many cases these designs are based on someone's thoughts about the way the system operates rather than on analysis. Simulation studies aid in providing understanding about how a system really operates rather than indicating someone's predictions about how a system will operate.

8. *Visualize the plan.* Taking your designs beyond CAD drwings by using the animation features offered by many simulation packages allows you to see your facility or organization actually running. Depending on the software used, you may be able to view your operations from various angles and levels of magnification, even in three dimensions. This allows you to detect design flaws that appear credible when seen just on paper on in a two-dimensional CAD drawing.

9. *Build consensus.* Using simulation to present design changes creates an objective opinion. You avoid having inferences made when you approve or disapprove of designs because you simply select the designs and modifications that provided the most desirable results, whether it be increased production or reducing the waiting time for service. In addition, it is much easier to accept reliable simulation results, which have been modeled, tested, validated, and visually represented, instead of one person's opinion of the results that will occur from a proposed design.

10. *Prepare for change.* We all know that the future will bring change. Answering all of the what-if questions is useful for both designing new systems and redesigning

existing systems. Interacting with all those involved in a project during the problem-formulation stage gives you an idea of the scenarios that are of interest. Then you construct the model so that it answers questions pertaining to those scenarios. What if an AGV is removed from service for an extended period of time? What if demand for service increases by 10%? What if ... ? The options are unlimited.

11. *Invest wisely.* The typical cost of a simulation study is substantially less than 1% of the total amount being expended for the implementation of a design or redesign. Since the cost of a change or modification to a system after installation is so great, simulation is a wise investment.

12. *Train the team.* Simulation models can provide excellent training when designed for that purpose. Used in this manner, the team provides decision inputs to the simulation model as it progresses. The team, and individual members of the team, can learn by their mistakes and learn to operate better. This is much less expensive and less disruptive than on-the-job learning.

13. *Specify requirements.* Simulation can be used to specify requirements for a system design. For example, the specifications for a particular type of machine in a complex system to achieve a desired goal may be unknown. By simulating different capabilities for the machine, the requirements can be established.

1.5.2 Disadvantages

The disadvantages of simulation include the following:

1. *Model building requires special training.* It is an art that is learned over time and through experience. Furthermore, if two models of the same system are constructed by two competent individuals, they may have similarities, but it is highly unlikely that they will be the same.

2. *Simulation results may be difficult to interpret.* Since most simulation outputs are essentially random variables (they are usually based on random inputs), it may be hard to determine whether an observation is a result of system interrelationships or randomness.

3. *Simulation modeling and analysis can be time consuming and expensive.* Skimping on resources for modeling and analysis may result in a simulation model and/or analysis that is not sufficient to the task.

4. *Simulation may be used inappropriately.* Simulation is used in some cases when an analytical solution is possible, or even preferable. This is particularly true in the simulation of some waiting lines where closed-form queueing models are available, at least for long-run evaluation.

In defense of simulation, these four disadvantages, respectively, can be offset as follows:

1. *Simulators.* Vendors of simulation software have been actively developing packages that contain models that only need input data for their operation. Such models have the generic tag "simulators" or templates.

2. *Output analysis.* Most simulation-software vendors have developed output-analysis capabilities within their packages or, as add on features for performing very extensive analysis. This reduces the computational requirements on the part of the user, although they still must understand the analysis procedure.

3. *Faster and faster.* Simulation can be performed faster today than yesterday, and even faster tomorrow. This is attributable to the advances in hardware that permit rapid running of scenarios. It is also attributable to the advances in many simulation packages. For example, many simulation software products contain constructs for modeling material handling using transporters such as conveyors and automated guided vehicles.

4. *Limitations of closed-form models.* Closed-form models are not able to analyze most of the complex systems that are encountered in practice. In nearly 8 years of consulting practice, not one problem has been encountered that could have been solved by a closed-form solution.

1.6 AREAS OF APPLICATION

The applications of simulation are vast. Recent presentations at the Winter Simulation Conference (WSC) can be divided into manufacturing, public systems, and service systems. WSC is an excellent way to learn more about the latest in simulation applications and theory. There are also numerous tutorials at both the beginning and advanced levels. WSC is sponsored by eight technical societies and the National Institute of Standards and Technology (NIST). The technical societies are the American Statistical Association (ASA), Association for Computing Machinery/Special Interest Group on Simulation (ACM/SIGSIM), Institute of Electrical and Electronics Engineers: Computer Society (IEEE/CS), Institute of Electrical and Electronics Engineers: Systems, Man and Cybernetics Society (IEEE/SMCS), Institute of Industrial Engineers (IIE), Institute for Operations Research and the Management Sciences, College on Simulation (INFORMS/CS), and Society for Computer Simulation (SCS). The societies can provide information about upcoming WSCs, usually held Monday through Wednesday in early December. Applications in the remainder of this section were presented at recent WSCs. (Chapter 25, in particular, contains references to recent Winter Simulation Conference Proceedings.) The major application areas of discrete-event simulation are discussed in Chapters 14 through 21.

1.6.1 Manufacturing and Material Handling Applications

Presentations included the following, among many others:

- Minimizing synchronization delays of prefabricated parts before assembly
- Evaluation of AGV routing strategies
- Flexible modeling and analysis of large-scale AS/RS-AGV systems
- Design and analysis of large-scale material handling systems
- Material flow analysis of automotive assembly plants
- Analysis of the effects of work-in-process levels on customer satisfaction
- Assessing the cost of quality

1.6.2 Public Systems Applications

Presentations included the following, among many others:

Health Systems

- Screening for abdominal aortic aneurysms
- Lymphocite development in immune-compromized patients
- Asthma dynamics and medical amelioration
- Timing of liver transplants
- Diabetic retinopathy
- Evaluation of nurse-staffing and patient-population scenarios
- Evaluation of automated equipment for a clinical processing laboratory
- Evaluation of hospital surgical suite and critical care area

Military Systems

- Air Force support equipment use
- Analysis of material handling equipment for prepositioning ships
- Development and implementation of measures of effectiveness
- Reengineering traditional stovepiped Army staffs for information operations
- Evaluation of theater airlift system productivity
- Evaluation of C-141 depot maintenance
- Evaluation of air mobility command channel cargo system

Natural Resources

- Nonpoint-source pollution analysis
- Weed scouting and weed control decision making
- Evaluation of surface water quality data

Public Services

- Emergency ambulance system analysis
- Evaluation of flow of civil lawsuits
- Evaluation of field offices within a government agency

1.6.3 Service System Applications

Presentations included the following, among many others:

Transportation

- Analysis of intelligent vehicle highway systems
- Evaluation of traffic control procedures at highway work zones
- Evaluation of taxi management and route control
- Animation of a toll plaza
- Port traffic planning model analysis
- Evaluation of rapid transit modeling with automatic and manual controls

Computer Systems Performance

- User transaction processing behavior analysis
- Evaluation of database transaction management protocols
- Evaluation of analytic models of memory queueing

Air Transportation

- Evaluation of human behavior in aircraft evacuations
- Analysis of airport/airline operations
- Evaluation of combination carrier air cargo hub

Communications Systems

- Trunked radio network analysis
- Evaluation of telephone service provisioning process
- Picture archiving and communications system analysis
- Evaluation of modeling of broadband telecommunication networks
- Analysis of virtual reality for telecommunication networks

1.7 STEPS IN A SIMULATION STUDY

Figure 1.1 shows a set of steps to guide a model builder in a thorough and sound simulation study. Similar figures and their interpretation can be found in other sources, such as Pegden et al. (1995) and Law and Kelton (1991). This presentation is built on that of Banks et al. (1996).

1. *Problem formulation.* Every simulation study begins with a statement of the problem. If the statement is provided by those that have the problem (client), the simulation analyst must take extreme care to ensure that the problem is clearly understood. If a problem statement is prepared by the simulation analyst, it is important that the client understand and agree with the formulation. It is suggested that a set of assumptions be prepared by the simulation analyst and agreed to by the client. Even with all of these precautions, it is possible that the problem will need to be reformulated as the simulation study progresses. This step is discussed further in Chapters 22 and 23.

2. *Setting of objectives and overall project plan.* Another way to state this step is "prepare a proposal." This step should be accomplished regardless of location of the analyst and client (i.e., as an external or internal consultant). The objectives indicate the questions that are to be answered by the simulation study. The project plan should include a statement of the various scenarios that will be investigated. The plans for the study should be indicated in terms of time that will be required, personnel that will be used, hardware and software requirements if the client wants to run the model and conduct the analysis, stages in the investigation, output at each stage, cost of the study and billing procedures, if any. This step is discussed further in Chapters 22 and 23.

3. *Model conceptualization.* The real-world system under investigation is abstracted by a conceptual model, a series of mathematical and logical relationships concerning

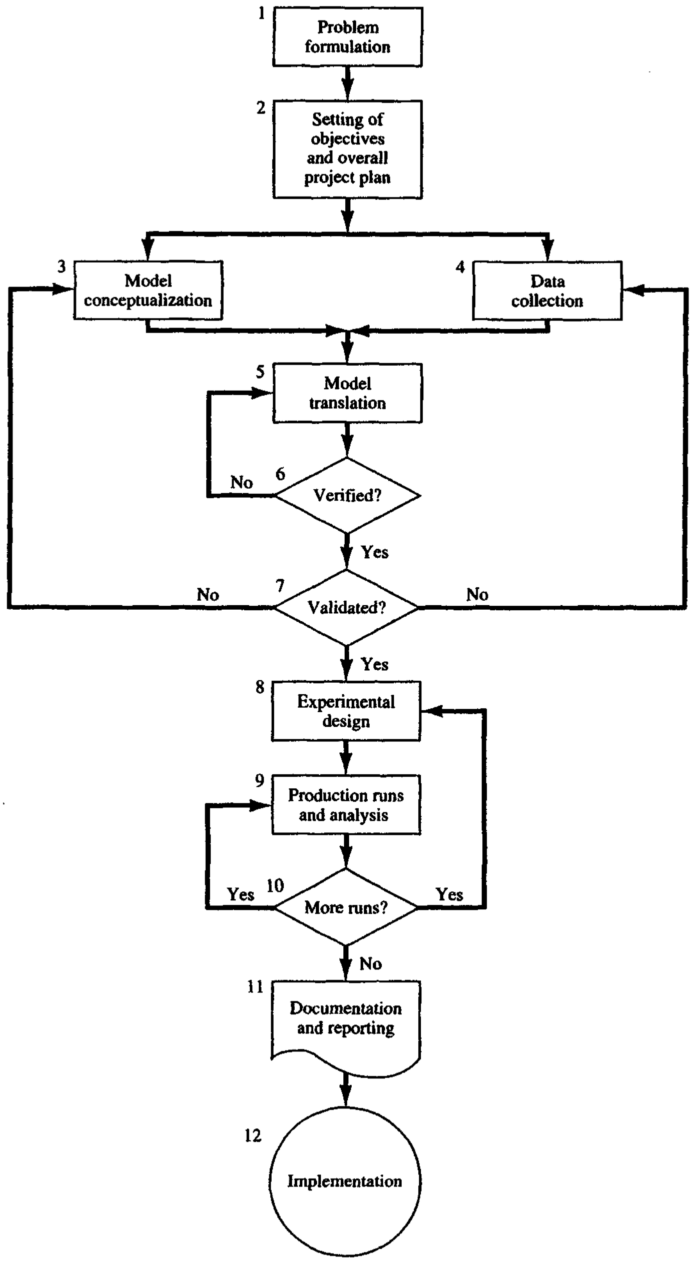

Figure 1.1 Steps in a simulation study. (From *Discrete-Event System Simulation*, 2nd ed., by Banks/Carson/Nelson, @ 1996. Reprinted by permission of Prentice Hall, Upper Saddle River, N.J.

the components and the structure of the system. It is recommended that modeling begin simply and that the model grow until a model of appropriate complexity has been developed. For example, consider the model of a manufacturing and material handling system. The basic model with the arrivals, queues, and servers is constructed. Then add the failures and shift schedules. Next, add the material-handling capabilities. Finally, add the special features. It is not necessary to construct an unduly complex model. This will add to the cost of the study and the time for its completion without increasing the quality of the output. The client should be involved throughout the model construction process. This will enhance the quality of the resulting model and increase the client's confidence in its use. This step is discussed further discussed in Chapters 2, 22, and 23.

4. *Data collection.* Shortly after the proposal is "accepted," a schedule of data requirements should be submitted to the client. In the best of circumstances, the client has been collecting the kind of data needed in the format required and can submit these data to the simulation analyst in electronic format. Often, the client indicates that the required data are indeed available. However, when the data are delivered they are found to be quite different than anticipated. For example, in the simulation of an airline-reservation system, the simulation analyst was told "we have every bit of data that you want over the last five years." When the study began the data delivered were the *average* "talk time" of the reservationist for each of the years. Individual values were needed, not summary measures. Model building and data collection are shown as contemporaneous in Figure 1.1. This is to indicate that the simulation analyst can readily construct the model while the data collection is progressing. This step is discussed further in Chapter 3.

5. *Model translation.* The conceptual model constructed in step 3 is coded into a computer-recognizable form, an operational model. This step is discussed further in Chapters 11, 12, 13, and 24.

6. *Verified?* Verification concerns the operational model. Is it performing properly? Even with small textbook-sized models, it is quite possible that they have verification difficulties. These models are orders of magnitude smaller than real models (say, 50 lines of computer code versus 2000 lines of computer code). It is highly advisable that verification take place as a continuing process. It is ill advised for the simulation analyst to wait until the entire model is complete to begin the verification process. Also, use of an interactive run controller, or debugger, is highly encouraged as an aid to the verification process. Verification is extremely important and is discussed further in this chapter. Additionally, this step is discussed extensively in Chapter 10.

7. *Validated?* Validation is the determination that the conceptual model is an accurate representation of the real system. Can the model be substituted for the real system for the purposes of experimentation? If there is an existing system, call it the base system, an ideal way to validate the model is to compare its output to that of the base system. Unfortunately, there is not always a base system (such as in the design of a new system). There are many methods for performing validation, and some of these are discussed further in this chapter. Additionally, this step is discussed extensively in Chapter 10.

8. *Experimental design.* For each scenario that is to be simulated, decisions need to be made concerning the length of the simulation run, the number of runs (also called *replications*), and the manner of initialization, as required. This step is discussed further in Chapter 6.

9. *Production runs and analysis.* Production runs, and their subsequent analysis, are used to estimate measures of performance for the scenarios that are being simulated. This step is discussed extensively in Chapters 7 to 9.

10. *More runs?* Based on the analysis of runs that have been completed, the simulation analyst determines if additional runs are needed and if any additional scenarios need to be simulated. This step is discussed extensively in Chapters 7 to 9.

11. *Documentation and reporting.* Documentation is necessary for numerous reasons. If the simulation model is going to be used again by the same or different analysts, it may be necessary to understand how the simulation model operates. This will stimulate confidence in the simulation model so that the client can make decisions based on the analysis. Also, if the model is to be modified, this can be greatly facilitated by adequate documentation. One experience with an inadequately documented model is usually enough to convince a simulation analyst of the necessity of this important step. The result of all the analysis should be reported clearly and concisely. This will enable the client to review the final formulation, the alternatives that were addressed, the criterion by which the alternative systems were compared, the results of the experiments, and analyst recommendations, if any. This step is discussed further in Chapters 22 and 23.

12. *Implementation.* The simulation analyst acts as a reporter rather than an advocate. The report prepared in step 11 stands on its merits and is just additional information that the client uses to make a decision. If the client has been involved throughout the study period, and the simulation analyst has followed all the steps rigorously, the likelihood of a successful implementation is increased. See Chapters 22 and 23 for more about implementation.

1.8 RANDOM NUMBER AND RANDOM VARIATE GENERATION

Example 1 used input values that were generated by a spinner and a die. Almost all simulation models are constructed within a computer, so spinners and dice are not devices that will be used. Instead, the computer will generate independent random numbers that are distributed continuously and uniformly between 0 and 1 [i.e., $U(0,1)$]. These random numbers can then be converted to the desired statistical distribution, or random variate, using one of several methods. Random variates are used to represent interarrival times, batch sizes, processing times, repair times, and time until failure, among others. Many researchers have written on the two topics in this section. These topics are discussed further in Chapters 4 and 5.

Simulation software products have a built-in random number generator (RNG) that produces a sequence of random numbers. Most of these generators are based on the linear congruential method (LCM), documented by Knuth (1969). A RNG is defined by its parameters, and some of them have been tested extensively. Chapter 4 introduces the topic of RNG.

The numbers generated by a RNG are actually pseudorandom. They are deterministic since they can be reproduced. Knowing the starting value, the values that follow it can be predicted, totally determining the sequence. There is no reason for concern since the length of the sequence prior to repeating itself is very, very long. On a 32-bit computer, this sequence can be longer than 2 billion. As reported in Chapter 4, even-longer-period RNGs are available.

The importance of a good source of random numbers is that all procedures for generating nonuniformly distributed random variates involve a mathematical transformation of uniform random numbers. For example, suppose that R_i is the ith random number generated from $U(0, 1)$. Suppose further that the desired random variate is exponentially distributed with rate λ. These values can be generated from

$$X_i = \frac{1}{\lambda} \ln(1 - R_i) \tag{1}$$

where X_i is the ith random variate generated [e.g., the time between the arrival of the ith and the $(i+1)$st entities]. Suppose that $\lambda = \frac{1}{10}$ arrival per minute. Using equation (1) [called the random variate generator (RVG)], if $R_1 = 0.3067$, then $X_1 = 3.66$ minutes. The RVG was developed using what is called the inverse-transform technique. Other techniques include convolution, acceptance–rejection, and composition. Techniques for RVG are discussed in Chapter 5.

Most simulation software products have built-in RVGs for the most widely used distributions and several that are not so widely utilized. The simulation software usually provides a facility for generating a sample from an empirical distribution (a distribution of the raw input data) that is either discrete or continuous. It is important that the simulation analyst know how to use RVGs, but it is not usually important to be concerned with their generation.

1.9 INPUT DATA

For each element in a system being modeled, the simulation analyst must decide on a way to represent the associated random variables. The presentation of the subject that follows is based on Banks et al. (1998). This topic is discussed in much more detail in Chapter 3.

The techniques used may vary depending on:

1. The amount of available data
2. Whether the data are "hard" or someone's best guess
3. Whether each variable is independent of other input random variables, or related in some way to other outputs

In the case of a variable that is independent of other variables, the choices are as follows:

1. Assume that the variable is deterministic.
2. Fit a probability distribution to the data.
3. Use the empirical distribution of the data.

These three choices are discussed in the next three subsections.

1.9.1 Assuming Randomness Away

Some simulation analysts may be tempted to assume that a variable is deterministic, or constant. This value could have been obtained by averaging historic information. The value may even be a guess. If there is randomness in the model, this technique can surely invalidate the results.

Suppose that a machine manufactures parts in exactly 1.5 minutes. The machine requires a tool change according to an exponential distribution with a mean of 12 minutes between occurrences. The tool change time is also exponentially distributed with a mean of 3 minutes. An inappropriate simplification would be to assume that the machine operates in a constant time of 1.875 minutes, and ignore the randomness. The consequences of these two interpretations are very great on such measures as the average number in the system or time waiting before the machine.

1.9.2 Fitting a Distribution to Data

If there are sufficient data points, say 50 or more, it may be appropriate to fit a probability distribution to the data using conventional methods. [Advanced methods for distribution fitting, such as that described by Wagner and Wilson (1993), are available to the interested reader.] When there are few data, the tests for goodness of fit offer little guidance in selecting one distribution form over another.

There are also underlying processes that give rise to distributions in a rather predictable manner. For example, if arrivals (1) occur one at a time, (2) are completely at random without rush or slack periods, and (3) are completely independent of one another, a Poisson process occurs. In such a case it can be shown that the number of arrivals in a given time period follows a Poisson distribution and the time between arrivals follows an exponential distribution.

Several vendors provide software to accomplish input data analysis. However, if a goodness-of-fit test is being conducted without the aid of input data analysis software, the following three-step procedure is recommended:

1. *Hypothesize a candidate distribution.* First, ascertain whether the underlying process is discrete or continuous. Discrete data arise from counting processes. Examples include the number of customers that arrive at a bank each hour, the number of tool changes in an 8-hour day, and so on. Continuous data arise from measurement (time, distance, weight, etc.). Examples include the time to produce each part and the time to failure of a machine. Discrete distributions frequently used in simulation include the Poisson, binomial, and geometric. Continuous distributions frequently used in simulation include the uniform, exponential, normal, triangular, lognormal, gamma, and Weibull. These distributions are described in virtually every engineering statistics text.

2. *Estimate the parameters of the hypothesized distribution.* For example, if the hypothesis is that the underlying data are normal, the parameters to be estimated from the data are the mean and the variance.

3. *Perform a goodness-of-fit test such as the chi-squared test.* If the test rejects the hypothesis, that is a strong indication that the hypothesis is not true. In that case, return to step 1, or use the empirical distribution of the data following the process described below.

The three-step procedure is described in engineering statistics texts and in many simulation texts, such as Banks et al. (1996) and Law and Kelton (1991). Even if software is being used to aid in the development of an underlying distribution, understanding the three-step procedure is recommended.

1.9.3 Empirical Distribution of the Data

When all possibilities have been exhausted for fitting a distribution using conventional techniques, the empirical distribution can be used. The empirical distribution uses the data as generated.

An example will help to clarify the discussion. The times to repair a conveyor system after a failure, denoted by x, for the previous 100 occurrences are given as follows:

Interval (hours)	Frequency of Occurrence
$0 < x \leq 1.0$	27
$1.0 < x \leq 2.0$	13
$2.0 < x \leq 3.0$	31
$3.0 < x \leq 4.0$	18
$4.0 < x \leq 8.0$	11

No distribution could be fit acceptably to the data using conventional techniques. It was decided to use the data as generated for the simulation. That is, samples were drawn, at random, from the continuous distribution shown above. This required linear interpolation so that simulated values might be in the form 2.89 hours, 1.63 hours, and so on.

1.9.4 When No Data Are Available

There are many cases where no data are available. This is particularly true in the early stages of a study, when the data are missing, when the data are too expensive to gather, or when the system being modeled is not in existence. One possibility in such a case is to obtain a subjective estimate, some call it a *guesstimate*, concerning the system. Thus if the estimate that the time to repair a machine is between 3 and 8 minutes, a crude assumption is that the data follow a uniform distribution with a minimum value of 3 minutes and a maximum value of 8 minutes. The uniform distribution is referred to as the *distribution of maximum ignorance* since it assumes that every value is equally likely. A better "guess" occurs if the "most likely" value can also be estimated. This would take the form "the time to repair the machine is between 3 and 8 minutes with a most likely time of 5 minutes." Now, a triangular distribution can be used with a minimum of 3 minutes, a maximum of 8 minutes, and a most likely value (mode) of 5 minutes.

As indicated previously, there are naturally occurring processes that give rise to distributions. For example, if the time to failure follows the (reasonable) assumptions of the Poisson process indicated previously, and the machine operator says that the machine fails about once every 2 hours of operation, an exponential distribution for time to failure could be assumed initially with a mean of 2 hours.

Estimates made on the basis of guesses and assumptions are strictly tentative. If, and when, data, or more data, become available, both the parameters and the distributional forms should be updated.

1.10 VERIFICATION AND VALIDATION

In the application of simulation, the real-world system under investigation is abstracted by a conceptual model. The conceptual model is then coded into the operational model. Hopefully, the operational model is an accurate representation of the real-world system. However, more than hope is required to ensure that the representation is accurate. There is a checking process that consists of two components:

1. *Verification:* a determination of whether the computer implementation of the conceptual model is correct. Does the operational model represent the conceptual model?
2. *Validation:* a determination of whether the conceptual model can be substituted for the real system for the purposes of experimentation.

The checking process is iterative. If there are discrepancies among the operational and conceptual models and the real-world system, the relevant operational model must be examined for errors, or the conceptual model must be modified to represent the real-world system better (with subsequent changes in the operational model). The verification and validation process should then be repeated. These two important topics are discussed extensively in Chapter 10.

1.10.1 Verification

The verification process involves examination of the simulation program to ensure that the operational model accurately reflects the conceptual model. There are many commonsense ways to perform verification.

1. *Follow the principles of structured programming.* The first principle is *top-down design* (i.e., construct a detailed plan of the simulation model before coding). The second principle is *program modularity* (i.e., break the simulation model into submodels). Write the simulation model in a logical, well-ordered manner. It is highly advisable (we would say mandatory if we could mandate such) to prepare a detailed flowchart indicating the macro activities that are to be accomplished. This is particularly true for real-world-sized problems. It is quite possible to think through all the computer code needed to solve problems at chapter ends of an academic text on discrete-event simulation. However, that computer code is minuscule compared to that of real-world problems.

2. *Make the operational model as self-documenting as possible.* This requires comments on virtually every line and sometimes between lines of code for those software products that allow programming. Imagine that one of your colleagues is trying to understand the computer code that you have written, but that you are not available to offer any explanation. For graphical software, on-screen documentation is suggested. In some cases, the text associated with documentation can be hidden from view when it is inappropriate to show it.

3. *Have the computer code checked by more than one person.* Several techniques have been used for this purpose. One of these can be called *code inspection.* There are four parties as follows: the moderator or leader of the inspection team, the designer or person who prepared the conceptual model, the coder or person who prepared the operational model, and the tester or the person given the verification responsibility. An

inspection meeting is held where a narration of the design is provided and the operational model is discussed, line by line, along with the documentation. Errors detected are documented and classified. There is then a rework phase, followed by another inspection. Alternatives to code inspection include the review, except that the interest is not line by line but in design deficiencies. Another alternative is the audit that verifies that the development of the computer code is proceeding logically. It verifies that the stated requirements are being met.

4. *Check to see that the values of the input data are being used appropriately.* For example, if the time unit is minutes, all of the data should be in terms of minutes, not hours or seconds.

5. *For a variety of input-data values, ensure that the outputs are reasonable.* Many simulation analysts are satisfied when they receive output. But that is far from enough. If there are 100 entities in a waiting line when 10 would be rather high, there is probably something wrong. For example, the resource actually has a capacity of two, but was modeled with a capacity of one.

6. *Use an interactive run controller (IRC) or debugger to check that the program operates as intended.* The IRC is a very important verification tool that should be used for all real-system models. An example of one of the capabilities of the IRC is the trace that permits following the execution of the model step by step.

7. *Animation is a very useful verification tool.* Using animation, the simulation analyst can detect actions that are illogical. For example, it may be observed that a resource is supposed to fail as indicated by turning red on the screen. While watching the animation, the resource never turned red. This could signal a logical error.

1.10.2 Validation

A variety of subjective and objective techniques can be used to validate the conceptual model. Sargent (1992) offers many suggestions for validation. Subjective techniques include the following:

1. *Face Validation.* A conceptual model of a real-world system must appear reasonable "on its face" to those who are knowledgeable (the "experts") about the real-world system. For example, the experts can validate that the model assumptions are correct. Such a critique by experts would aid in identifying deficiencies or errors in the conceptual model. The credibility of the conceptual model would be enhanced as these deficiencies or errors are eliminated.

2. *Sensitivity Analysis.* As model input is changed, the output should change in a predictable direction. For example, if the arrival rate increases, the time in queues should increase, subject to some exceptions. (An example of an exception is as follows: If a queue increases, it may be the case that resources are added within the model, negating the prediction.)

3. *Extreme-Condition Tests.* Does the model behave properly when input data are at the extremes? If the arrival rate is set extremely high, does the output reflect this change with increased numbers in the queues, increased time in the system, and so on?

4. *Validation of Conceptual Model Assumptions.* There are two types of conceptual model assumptions. They are structural assumptions (concerning the operation of the real-world system) and data assumptions. Structural assumptions can be validated by

observing the real-world system and by discussing the system with the appropriate personnel. No one person knows everything about the entire system. Many people need to be consulted to validate conceptual model assumptions.

Information from intermediaries should be questioned. A simulation consulting firm often works through other consulting firms. An extremely large model of a distant port operation was constructed. It was only after a visit by the simulation consulting firm to the port that it was discovered that one of the major model assumptions concerning how piles of iron ore are formed was in error.

Assumptions about data should also be validated. Suppose it is assumed that times between arrivals of customers to a bank during peak periods are independent and in accordance with an exponential distribution. To validate conceptual model assumptions, the following would be in order:

(a) Consult with appropriate personnel to determine when peak periods occur.
(b) Collect interarrival data from these periods.
(c) Conduct statistical tests to ensure that the assumption of independence is reasonable.
(d) Estimate the parameter of the assumed exponential distribution.
(e) Conduct a goodness-of-fit test to ensure that the exponential distribution is reasonable.

5. *Consistency Checks.* Continue to examine the operational model over time. An example explains this validation procedure. A simulation model is used annually. Before using this model, make sure that there are no changes in the real system that must be reflected in the structural model. Similarly, the data should be validated. For example, a faster machine may have been installed in the interim period, but it was not included in the information provided.

6. *Turing Tests.* Persons knowledgeable about system behavior can be used to compare model output to system output. For example, suppose that five reports of actual system performance over five different days are prepared and five simulated outputs are generated. These 10 reports should be in the same format. The 10 reports are randomly shuffled and given to a person, say an engineer, who has seen this type of information. The engineer is asked to distinguish between the two kinds of reports, actual and simulated. If the engineer identifies a substantial number of simulated reports, the model is inadequate. If the engineer cannot distinguish between the two, there is less reason to doubt the adequacy of the model.

Objective techniques include the following:

7. *Validating Input–Output Transformations.* The basic principle of this technique is the comparison of output from the operational model to data from the real system. Input–output validation requires that the real system currently exist. One method of comparison uses the familiar t-test, discussed in most statistics texts.

8. *Validation Using Historical Input Data.* Instead of running the operational model with artificial input data, we could drive the operational model with the actual historical record. It is reasonable to expect the simulation to yield output results within acceptable statistical error of those observed from the real-world system. The paired t-test,

discussed in most statistics texts, is one method for conducting this type of validation.

1.11 EXPERIMENTATION AND OUTPUT ANALYSIS

The analysis of simulation output begins with the selection of performance measures. Performance measures can be time weighted, based on counting of occurrences, or arise from the tabulation of expressions including means, variances, and so on.

An example of a time-weighted statistic is the average number in system over a time period of length T. Figure 1.2 shows the number in system, $L(t)$, at time t, from $t = 0$ to $t = 60$. The time-weighted average number in the system, $\bar{L}$, at $T = 60$ is given by the sum of the areas of the rectangles divided by T. Thus

$$\bar{L} = \frac{(0 \times 10) + (1 \times 10) + (2 \times 15) + (1 \times 10) + (0 \times 5) + (1 \times 6) + (2 \times 4)}{60}$$

$$= 1.07$$

An example of a statistic based on counting of occurrences is the number of acceptable units completed in 24 hours of simulated time. A statistic based on the tabulation of expressions is the patent royalties from three different part types, each with a different contribution per unit, for a 24-hour period.

The simulation of a stochastic system results in performance measures that contain random variation. Proper analysis of the output is required to obtain sound statistical results from these replications. Specific questions that must be addressed when conducting output analysis are:

1. What is the appropriate run length of the simulation (unless the system dictates a value)?
2. How do we interpret the simulated results?
3. How do we analyze the differences between different model configurations?

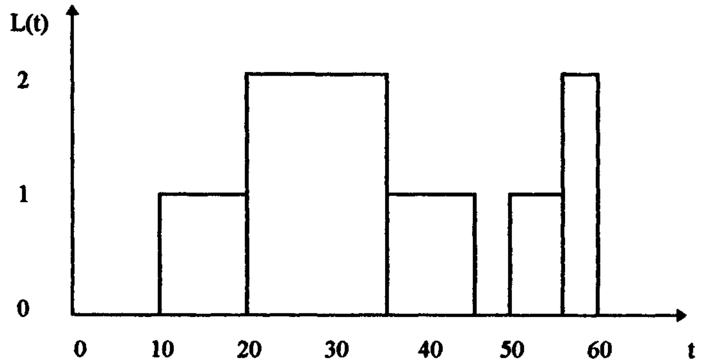

Figure 1.2 Number in system, $L(t)$, at time t.

These topics are introduced in the next section. They are discussed extensively in Chapters 6 to 8.

1.11.1 Statistical Confidence and Run Length

A confidence interval for the performance measure being estimated by the simulation model is a basic component of output analysis. A confidence interval is a numerical range that has a probability of $1-\alpha$ of including the true value of the performance measure, where $1-\alpha$ is the confidence level for the interval. For example, let us say that the performance measure of interest is the mean time in the queue, μ, and a $100(1 - \alpha)\%$ confidence interval for μ is desired. If many replications are performed and independent confidence intervals on μ are constructed from those replications, approximately $100(1 - \alpha)\%$ of those intervals will contain μ. Consider the following example.

Example 2 (Confidence Intervals) Given the data in Table 1.2, both a 95% (α = 0.05) and a 99% (α = 0.01) two-sided confidence interval are desired. Assuming that the values for X are normally distributed, a $1 - \alpha$ confidence interval for the mean, μ, is given by $(\overline{X} - h, \overline{X} + h)$, where $\overline{X}$ is the sample mean and h is the half-width.
The equation for $\overline{X}$ is given by

$$\overline{X} = \sum_{i=1}^{n} \frac{X_i}{n} \tag{2}$$

The half-width h of the confidence interval is computed as follows:

$$h = t_{n-1,1-\alpha/2} \frac{S}{\sqrt{n}} \tag{3}$$

where $t_{n-1,1-\alpha/2}$ is the upper $1 - \alpha/2$ critical value of the t-distribution with $n - 1$ degrees of freedom, and S is the sample standard deviation. To compute S, first use equation (4) to compute S^2 as follows:

$$S^2 = \frac{\sum_{i=1}^{n} X_i^2 - n\overline{X}^2}{n-1} \tag{4}$$

Table 1.2 Data for Example 2

Replication Number, i	Average Time in Queue, X_i
1	63.2
2	69.7
3	67.3
4	64.8
5	72.0

Taking the square root of S^2 yields S.

Since a two-sided confidence interval is desired, we use $\alpha/2$ to compute the half-width. Using equations (2) and (4), we obtain $\overline{X} = 67.4$ and $S = 3.57$. In addition,

$$t_{4,.975} = 2.78 \quad (95\% \text{ confidence})$$
$$t_{4,.995} = 4.60 \quad (99\% \text{ confidence})$$

resulting in

$$h = \begin{cases} 4.44 & (95\% \text{ confidence}) \\ 7.34 & (99\% \text{ confidence}) \end{cases}$$

The confidence interval is given by $(\overline{X} - h, \overline{X} + h)$. Therefore, the 95% confidence interval is (62.96, 71.84), and the 99% confidence interval is (60.06, 74.74).

As demonstrated in Example 2, the size of the interval depends on the confidence level desired, the sample size, and the inherent variation (measured by S). The higher level of confidence (99%) requires a larger interval than the lower confidence level (95%). In addition, the number of replications, n, and their standard deviation, S, are used in calculating the confidence interval. In simulation, each replication is considered one data point. Therefore, the three factors that influence the width of the confidence interval are:

1. Number of replications (n)
2. Level of confidence ($1 - \alpha$)
3. Variation of performance measure (S)

The relationship between these factors and the confidence interval is:

1. As the number of replications increases, the width of the confidence interval decreases.
2. As the level of confidence increases, the width of the interval increases. In other words, a 99% confidence interval is larger than the corresponding 95% confidence interval.
3. As the variation increases, the width of the interval increases.

1.11.2 Terminating Versus Nonterminating Systems

The procedure for output analysis differs based on whether the system is terminating or nonterminating. In a terminating system, the duration of the simulation is fixed as a natural consequence of the model and its assumptions. The duration can be fixed by specifying a finite length of time to simulate or by limiting the number of entities created or disposed. An example of a terminating system is a bank that opens at 9:00 A.M. and closes at 4:00 P.M. Some other examples of terminating systems include a check-processing facility that operates from 8:00 P.M. until all checks are processed, a ticket booth that remains open until all the tickets are sold or the event begins, and a manufacturing facility that processes a fixed number of jobs each day and then shuts down.

By definition, a terminating system is one that has a fixed starting condition and an event definition that marks the end of the simulation. The system returns to the fixed initial condition, usually "empty and idle," before the system begins operation again. The objective of the simulation of terminating systems is to understand system behavior for a "typical" fixed duration. Since the initial starting conditions and the length of the simulation are fixed, the only controllable factor is the number of replications.

One analysis procedure for terminating systems is to simulate a number of replications, compute the sample variance of the selected estimator measure, and determine if the width of the resulting confidence interval is within acceptable limits. For example, if the average number of parts in the queue is of interest, the first step is to conduct a pilot run of n replications. Next, compute the confidence interval for the expected average number of parts in the queue using the observations recorded from each replication. Then if the confidence interval is too large, determine the number of additional replications required to bring it within limits. Finally, conduct the approximate additional replications and recompute the new confidence interval using all the data. Iterate the last two steps until the confidence interval is of satisfactory size.

In a nonterminating system, the duration is not finite; the system is in perpetual operation. An example of a nonterminating system is an assembly line that operates 24 hours a day, 7 days a week. Another example of this type of system is the manufacture of glass fiber insulation for attics. If operation of the system is stopped, the molten glass will solidify in the furnace, requiring that it be chipped away tediously before restarting the system. The objective in simulating a nonterminating system is to understand the long-run, or *steady-state*, behavior. To study steady-state behavior accurately, the effects of the initial conditions, or transient phase, must be removed from the simulation results. This can be accomplished by swamping, preloading, or deletion.

The first method, *swamping*, suppresses the initial-condition effects by conducting a very long simulation run, so long that any initial conditions have only a minuscule effect on the long-run value of the performance measure. For example, if the initial conditions last for 100 hours, simulate for 10,000 hours. A problem with the swamping technique is that the bias from starting empty and idle will always exist, even if it is small.

The second method, *preloading*, primes the system before simulation starts by placing entities in delay blocks and queues. In other words, an attempt is made to have the initial conditions match the steady-state conditions. This requires some rough knowledge of how the system looks in steady state. Thus, if we are simulating a bank that has one line forming before three tellers, we need to observe the bank in operation to obtain information about the usual situation. For example, we may find that the three tellers are usually busy and that there are about four people in line. This is how the simulation would begin when using the preloading technique. The bank is a very simple system to observe. However, for more complex systems, this initialization procedure becomes somewhat difficult, especially if the system is still in the design phase.

The third method, *deletion*, excludes the initial transient phase that is influenced by the initial conditions. Data are collected from the simulation only after the transient (warm-up) phase has ended. This idea is demonstrated in Figure 1.3. The difficulty with the deletion method is the determination of the length of the transient phase. Although elegant statistical techniques have been developed, a satisfactory method is to plot the

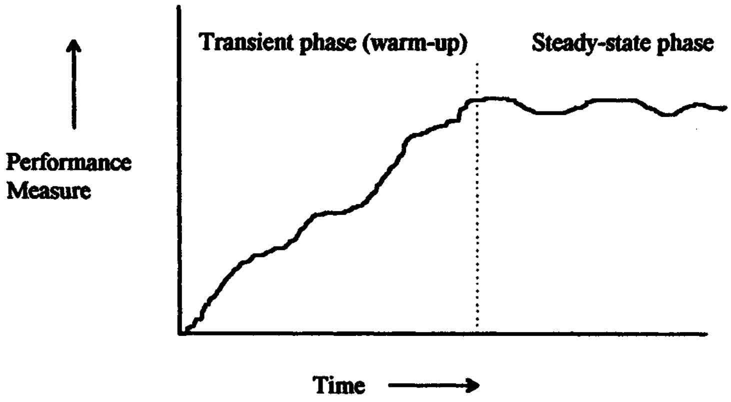

Fig. 1.3 Deletion of initial observations for a nonterminating system.

output of interest over time and visually observe when steady state is reached. Welch (1983) provides a formal description of this method.

1.12 SUMMARY

The chapter began with a definition of simulation, including an example. Underlying concepts were presented, including the system and model, system state variables, entities and attributes, list processing, activities and delays, and the definition of discrete-event simulation. Next, four modeling structures were discussed, including process interaction, event scheduling, activity scanning, and the three-phase method. The advantages and disadvantages of simulation were presented, with amelioration of the disadvantages. Next, areas of application from presentations at the Winter Simulation Conference were shown. The steps in a simulation study were given with a brief discussion of each. The manner in which random numbers and random variates are generated was presented next. Three ways that might be used for generating input data were described. However, the first method, assuming randomness away, is discouraged. The extremely important concepts of verification and validation were then discussed. The all-important topic of experimentation and output analysis was introduced. The topics introduced in this chapter are discussed much more extensively in the remaining chapters.

REFERENCES

Balci, O. (1988). The implementation of four conceptual frameworks for simulation modeling in high-level languages, in *Proceedings of the 1988 Winter Simulation Conference*, M. Abrams, P. Haigh, and J. Comfort, eds., IEEE, Piscataway, N.J.

Banks, J., and V. Norman (1995). Justifying simulation in today's manufacturing environment, *IIE Solutions*, November.

Banks, J., B. Burnette, H. Kozloski, and J. D. Rose (1995). *Introduction to SIMAN V and CINEMA V*, Wiley, New York.

Banks, J., J. S. Carson II, and B. L. Nelson (1996). *Discrete-Event System Simulation*, 2nd ed., Prentice Hall, Upper Saddle River, N.J.

Banks, J., J. S. Carson II, and D. Goldsman (1998). Discrete-event computer simulation, in *Handbook of Statistical Methods for Engineers and Scientists*, 2nd ed., H. M. Wadsworth, ed., McGraw-Hill, New York.

Carson, J. S. (1993). Modeling and simulation world views, in *Proceedings of the 1993 Winter Simulation Conference*, G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, eds., IEEE, Piscataway, N.J., pp. 18–23.

Knuth, D. W. (1969). *The Art of Computer Programming*, Vol. 2: *Semi-numerical Algorithms*, Addison-Wesley, Reading, Mass.

Law, A. M., and W. D. Kelton (1991). *Simulation Modeling and Analysis*, 2nd ed., McGraw-Hill, New York.

Pegden, C. D., R. E. Shannon, and R. P. Sadowski (1995). *Introduction to Simulation Using SIMAN*, 2nd ed., McGraw-Hill, New York.

Pidd, M. (1992). *Computer Modelling for Discrete Simulation*, Wiley, Chichester, West Sussex, England.

Sargent, R. G. (1992). Validation and verification of simulation models, in *Proceedings of the 1992 Winter Simulation Conference*, J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson, eds., IEEE, Piscataway, N.J., pp. 104–114.

Schriber, T. J. (1991). *An Introduction to Simulation Using GPSS/H*, Wiley, New York.

Wagner, M. A. F., and J. R. Wilson (1993). Using univariate Bézier distributions to model simulation input processes, in *Proceedings of the 1993 Winter Simulation Conference*, G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, eds., IEEE, Piscataway, N.J., pp. 365–373.

Welch, P. D. (1983). The statistical analysis of simulation results, in *The Computer Performance Modeling Handbook*, S. Lavenberg, ed., Academic Press, San Diego, Calif.