4.22. For the confidence interval given by (4.12), show that the coverage approaches $1 - \alpha$ as $n \to \infty$.

4.23. Suppose that 7.3, 6.1, 3.8, 8.4, 6.9, 7.1, 5.3, 8.2, 4.9, and 5.8 are 10 observations from a distribution (not highly skewed) with unknown mean μ. Compute $\bar{X}(10)$, $S^2(10)$, and an approximate 95 percent confidence interval for μ.

4.24. For the data in Prob. 4.23, test the null hypothesis H_0: $\mu = 6$ at level $\alpha = 0.05$.

4.25. Suppose that X and Y are random variables with unknown covariance $\text{Cov}(X, Y)$. If the pairs X_i, Y_i (for $i = 1, 2, \ldots, n$) are independent observations of X, Y, then show that

$$\widehat{\text{Cov}}(X, Y) = \frac{\sum_{i=1}^{n} [X_i - \bar{X}(n)][Y_i - \bar{Y}(n)]}{n - 1}$$

is an unbiased estimator of $\text{Cov}(X, Y)$.

4.26. A random variable X is said to have the *memoryless property* if

$$P(X > t + s \mid X > t) = P(X > s) \qquad \text{for all } t, s > 0$$

[The conditional probability $P(X > t + s \mid X > t)$ is the probability of the event $\{X > t + s\}$ occurring given that the event $\{X > t\}$ has occurred; see Ross (2003, chap. 3).] Show that the exponential distribution has the memoryless property.

4.27. A geometric distribution with parameter p $(0 < p < 1)$ has probability mass function

$$p(x) = p(1 - p)^x \qquad \text{for } x = 0, 1, 2, \ldots$$

Show that this distribution has the memoryless property.

4.28. Suppose that a man has k keys, one of which will open a door. Compute the expected number of keys required to open the door for the following two cases:

(*a*) The keys are tried one at a time without replacement.

(*b*) The keys are tried one at a time with replacement. (*Hint:* Condition on the outcome of the first try.)

4.29. Are the mean, median, and mode equal for every symmetric distribution?

CHAPTER 5

Building Valid, Credible, and Appropriately Detailed Simulation Models

Recommended sections for a first reading: 5.1 through 5.5, 5.6.1

5.1 INTRODUCTION AND DEFINITIONS

One of the most difficult problems facing a simulation analyst is that of trying to determine whether a simulation model is an accurate representation of the actual system being studied, i.e., whether the model is *valid*. In this chapter we present a *practical* discussion of how to build valid and credible models. We also provide guidelines on how to determine the level of detail for a model of a complex system, also a critical and challenging issue. Information for this chapter came from a review of the existing literature, from consulting studies performed by Averill M. Law & Associates, and from the experiences of the thousands of people who have attended the author's simulation short courses since 1977. We present more than 40 examples to illustrate the concepts presented.

Important works on validation and verification include those by Balci (1998), Banks et al. (2005), Carson (1986, 2002), Feltner and Weiner (1985), Law (2000, 2005), Naylor and Finger (1967), Sargent (2004), Shannon (1975), and Van Horn (1971). References on the assessment of an existing simulation model include Fossett et al. (1991), Gass (1983), Gass and Thompson (1980), and Knepell and Arangno (1993).

We begin by defining the important terms used in this chapter, including verification, validation, and credibility. *Verification* is concerned with determining whether the "assumptions document" (see Sec. 5.4.3) has been correctly translated into a computer "program," i.e., debugging the simulation computer program. Although verification is simple in concept, debugging a large-scale simulation program is a difficult and arduous task due to the potentially large number of logical paths. Techniques for verifying a simulation computer program are discussed in Sec. 5.3.

Validation is the *process* of determining whether a simulation model is an accurate representation of the system, *for the particular objectives of the study*. [Fishman and Kiviat (1968) appear to be the first ones to have given definitions similar to these.] The following are some general perspectives on validation:

- Conceptually, if a simulation model is "valid," then it can be used to make decisions about the system similar to those that would be made if it *were* feasible and cost-effective to experiment with the system itself.
- The ease or difficulty of the validation process depends on the complexity of the system being modeled and on *whether a version of the system currently exists* (see Sec. 5.4.5). For example, a model of a neighborhood bank would be relatively easy to validate since it could be closely observed. However, a model of the effectiveness of a naval weapons system in the year 2025 would be impossible to validate completely, since the location of the battle and the nature of the enemy weapons would be unknown.
- A simulation model of a complex system can only be an *approximation* to the actual system, no matter how much effort is spent on model building. There is no such thing as absolute model validity, nor is it even desired. The more time (and hence money) that is spent on model development, the more valid the model should be in general. However, the most valid model is not necessarily the most cost-effective. For example, increasing the validity of a model beyond a certain level might be quite expensive, since extensive data collection may be required, but might not lead to significantly better insight or decisions.
- A simulation model should always be developed for a particular set of purposes. Indeed, a model that is valid for one purpose may not be for another.
- The measures of performance used to validate a model should include those that the decision maker will actually use for evaluating system designs.
- Validation is not something to be attempted after the simulation model has already been developed, and only if there is time and money remaining. Unfortunately, our experience indicates that this recommendation is often not followed.

EXAMPLE 5.1. An organization paid a consulting company \$500,000 to perform a "simulation study." After the study was supposedly completed, a person from the client organization called and asked, "Can you tell me in *five minutes* on the phone how to validate our model?"

- Each time a simulation model is being considered for a new application, its validity should be reexamined. The current purpose may be substantially different from the original purpose, or the passage of time may have invalidated certain model parameters.

A simulation model and its results have *credibility* if the manager and other key project personnel accept them as "correct." (We will henceforth use the term "manager" to mean manager, decision maker, or client, as is appropriate to the context.) Note that a credible model is not necessarily valid, and vice versa. Also, a model can be credible and not actually used as an aid in making decisions. For

example, a model could be credible but not used because of political or economic reasons. The following things help establish credibility for a model:

- The manager's understanding of and agreement with the model's assumptions (see Sec. 5.4.2)
- Demonstration that the model has been validated and verified
- The manager's ownership of and involvement with the project
- Reputation of the model developers
- A compelling animation

The U.S. Department of Defense (DoD) is a large user of simulation models, and in recent years there has been considerable interest in verification, validation, and a concept known as accreditation (VV&A). *Accreditation* [see Defense Modeling and Simulation Office (2000, 2003)] is the official certification (by the project sponsor) that a simulation model is acceptable for a specific purpose. The main reason that accreditation is mandated within DoD is that someone must take responsibility for the decision to use a model for a particular application, since a large amount of money and people's lives may be at stake. Also, most military analyses are done with legacy models, which may have been developed for another application or by another organization. Issues that are considered in an accreditation decision include:

- Verification and validation that have been done
- Credibility of the model
- Simulation model development and use history (e.g., model developer and similar applications)
- Quality of the data that are available
- Quality of the documentation
- Known problems or limitations with the simulation model

The timing and relationships of validation, verification, and establishing credibility are shown in Fig. 5.1. The rectangles represent states of the model or the system of interest, the solid horizontal arrows correspond to the actions necessary to move from one state to another, and the curved dashed arrows show where the three major concepts are most prominently employed. The numbers below each solid

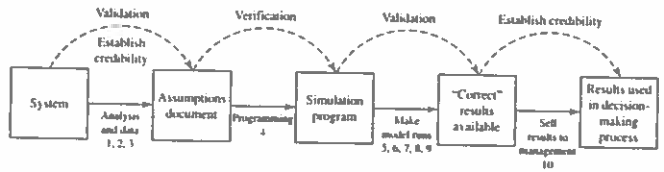

FIGURE 5.1
Timing and relationships of validation, verification, and establishing credibility.

arrow correspond to the steps in a sound simulation study, as discussed in Sec. 1.7. We have not attempted to illustrate feedback arcs in the figure.

Validation should be contrasted with *output analysis* (the subject of Chaps. 9 through 12), which is a statistical issue concerned with estimating a simulation *model's* (not necessarily the system's) true measures of performance. Topics of interest in output analysis include simulation run length, length of the warmup period (if any), and number of independent model runs (using different random numbers).

To get a better idea of the difference between validation and output analysis, suppose that we want to estimate the mean μ_S of some system. Suppose that we construct a simulation model whose corresponding mean is μ_M. We make a simulation run and obtain an estimate $\hat{\mu}_M$ of μ_M. Then the error in $\hat{\mu}_M$ as an estimate of μ_S is given by

$$
\begin{aligned}
\text{Error in } \hat{\mu}_M &= |\hat{\mu}_M - \mu_S| \\
&= |\hat{\mu}_M - \mu_M + \mu_M - \mu_S| \\
&\le |\hat{\mu}_M - \mu_M| + |\mu_M - \mu_S| \qquad \text{(by the triangle inequality)}
\end{aligned}
$$

Validation is concerned with making the second absolute value small (in the line above), while output analysis is concerned with making the first absolute value small. Thus, to have a good estimate of the mean of the system, we have to be concerned with *both* validation and output analysis.

5.2 GUIDELINES FOR DETERMINING THE LEVEL OF MODEL DETAIL

A simulation practitioner must determine what aspects of a complex real-world system actually need to be incorporated into the simulation model and at what level of detail, and what aspects can be safely ignored. It is rarely necessary to have a one-to-one correspondence between each element of the system and each element of the model. Modeling each aspect of the system will seldom be required to make effective decisions, and might result in excessive model execution time, in missed deadlines, or in obscuring important system factors.

EXAMPLE 5.2. A dog-food manufacturer had a consulting company build a simulation model of its manufacturing line, which produced 1 million cans per day at a constant rate. Because each can of food was represented by a separate entity in the model, the model was very expensive to run and, thus, not very useful. A few years later the model was rewritten, treating the manufacturing process as a "continuous flow" (see Sec. 1.2). The new model produced accurate results and executed in a small fraction of the time necessary for the original model.

EXAMPLE 5.3. A simulation model of a 1.5-mile-long factory was built in 1985 at a cost of $250,000. However, the model was so detailed that no runs were ever made due to excessive computer memory requirements.

We now present some general guidelines for determining the level of detail required by a simulation model [see also Law (1991) and Robinson (2004, pp. 87–92)].

- Carefully define the *specific* issues to be investigated by the study and the measures of performance that will be used for evaluation. Models are not universally valid, but are designed for specific purposes. If the issues of interest have not been delineated, then it is impossible to determine the appropriate level of model detail. Since some models can accurately estimate one measure of performance but not another, it is also important to specify the performance measures of interest. For example, a simple model of a manufacturing system might accurately predict throughput (e.g., parts per day) but be inadequate for determining the required floor space for work-in-process (see Example 13.3). Finally, it is important to understand the manager's needs. A great model for the wrong problem will never be used. Problem formulation is usually done at an initial kickoff meeting with people representing all key aspects of the system being present.

EXAMPLE 5.4. A U.S. military analyst worked on a simulation model for six months without interacting with the general who requested it. At the Pentagon briefing for the study, the general walked out after 5 minutes stating, "That's not the problem I'm interested in."

- The entity moving through the simulation model does not always have to be the same as the entity moving through the corresponding system (see Example 5.5). Furthermore, it is not always necessary to model each component of the system in complete detail (see Example 5.26).

EXAMPLE 5.5. A large food manufacturer built a simulation model of its manufacturing line for snack crackers. Initially, they tried to model each cracker as a separate entity, but the computational requirements of the model made this approach infeasible. As a result, the company was forced to use a box of crackers as the entity moving through the model. The validity of this modeling approach was determined by using sensitivity analysis (see below and Example 5.25).

- Use subject-matter experts (SMEs) and sensitivity analyses to help determine the level of model detail. People who are familiar with systems similar to the one of interest are asked what components of the proposed system are likely to be the most important and, thus, need to be carefully modeled. Sensitivity analyses (see Sec. 5.4.4) can be used to determine what system factors (e.g., parameters or distributions) have the greatest impact on the desired measures of performance. Given a limited amount of time for model development, one should obviously concentrate on the most important factors.
- A mistake often made by beginning modelers is to include an excessive amount of model detail. As a result, we recommend starting with a "moderately detailed" model, which can later be embellished if needed. The adequacy of a particular version of the model is determined in part by presenting the model to SMEs and managers. Regular interaction with these people also maintains their interest in the simulation study.

EXAMPLE 5.6. We developed a simulation model of a pet-food manufacturing system, which consisted of a meat plant and a cannery. In the meat plant, meat was either ground fine or into chunks and then placed into buckets and transported to the cannery by an overhead conveyor system. In the cannery buckets are dumped into mixers that process the meat and then dispense it to filler/seamers for canning. The empty buckets are conveyed back to the meat plant for refilling. Originally, it was decided that the system producing the chunky product was relatively unimportant and, thus, it was modeled in a simple manner. However, at the structured walk-through of the model (see Sec. 5.4.3), machine operators stated that this subsystem was actually much more complex. To gain credibility with these members of the project team, we had to include machine breakdowns and contention for resources. Furthermore, after the initial model runs were made, it was necessary to make additional changes to the model suggested by a mixer operator.

- Do not have more detail in the model than is necessary to address the issues of interest, subject to the proviso that the model must have enough detail to be credible. Thus, it may sometimes be necessary to include things in a model that are not strictly required for model validity, due to credibility concerns.
- The level of model detail should be consistent with the type of data available. A model used to design a new manufacturing system will generally be less detailed than one used to fine-tune an existing system, since little or no data will be available for a proposed system.
- In virtually all simulation studies, time and money constraints are a major factor in determining the amount of model detail.
- If the number of factors (aspects of interest) for the study is large, then use a "coarse" simulation model or an analytic model to identify what factors have a significant impact on system performance. A "detailed" simulation model is then built, emphasizing these factors [see Haider, Noller, and Robey (1986) for an example]. Note that there are commercial software packages available for performing analytic analyses in application areas such as manufacturing systems and communications networks. Statistical experimental design (see Chap. 12) might also be useful for determining important factors.

5.3 VERIFICATION OF SIMULATION COMPUTER PROGRAMS

In this section we discuss eight techniques that can be used to debug the computer program of a simulation model [see Balci (1998) for additional techniques from the field of software engineering]. Some of these techniques may be used to debug any computer program, while others we believe to be unique to simulation modeling.

Technique 1

In developing a simulation model, write and debug the computer program in modules or subprograms. By way of example, for a 10,000-statement simulation model it would be poor programming practice to write the entire program before attempting any debugging. When this large, untested program is finally run, it almost certainly will not execute, and determining the location of the errors in the program will be extremely difficult. Instead, the simulation model's main program and a few of the key subprograms should be written and debugged first, perhaps representing the other required subprograms as "dummies" or "stubs." Next, additional subprograms or levels of detail should be added and debugged successively, until a model is developed that satisfactorily represents the system under study. In general, it is always better to start with a "moderately detailed" model, which is gradually made as complex as needed, than to develop "immediately" a complex model, which may turn out to be more detailed than necessary and excessively expensive to run (see Example 5.25 for further discussion).

EXAMPLE 5.7. For the multiteller bank with jockeying considered in Sec. 2.6, a good programming approach would be first to write and debug the computer program without letting customers jockey from queue to queue.

Technique 2

It is advisable in developing large simulation models to have more than one person review the computer program, since the writer of a particular subprogram may get into a mental rut and, thus, may not be a good critic. In some organizations, this idea is implemented formally and is called a *structured walk-through of the program*. For example, all members of the modeling team, say, systems analysts, programmers, etc., are assembled in a room, and each is given a copy of a particular set of subprograms to be debugged. Then the subprograms' developer goes through the programs but does not proceed from one statement to another until everyone is convinced that a statement is correct.

Technique 3

Run the simulation under a variety of settings of the input parameters, and check to see that the output is reasonable. In some cases, certain simple measures of performance may be computed exactly and used for comparison. (See the case study in Sec. 13.6.)

EXAMPLE 5.8. For many queueing systems with s servers in parallel, it can be shown that the long-run average utilization of the servers is $\rho = \lambda/(s\omega)$ (see App. 1B for notation). Thus, if the average utilization from a simulation run is close to the utilization factor ρ, there is some indication that the program may be working correctly.

Technique 4

One of the most powerful techniques that can be used to debug a discrete-event simulation program is a "trace." In a *trace*, the state of the simulated system, i.e., the contents of the event list, the state variables, certain statistical counters, etc., are displayed just after each event occurs and are compared with hand calculations to see if the program is operating as intended. In performing a trace it is desirable to evaluate each possible program path as well as the program's ability to deal with "extreme" conditions. Sometimes such a thorough evaluation may require that special (perhaps deterministic) input data be prepared for the model. Most simulation packages provide the capability to perform traces.

A batch-mode trace often produces a large volume of output, which must be checked event by event for errors. Unfortunately, some key information may be omitted from the trace (not having been requested by the analyst); or, worse yet, a particular error may not occur in the "short" debugging simulation run. Either difficulty will require that the simulation be rerun. As a result, it is usually preferable to use an interactive debugger to find programming errors.

An *interactive debugger* allows an analyst to stop the simulation at a selected point in time, and to examine and possibly change the values of certain variables. This latter capability can be used to "force" the occurrence of certain types of errors. Many modern simulation packages have an interactive debugger.

EXAMPLE 5.9. Table 5.1 shows a trace for the intuitive explanation of the single-server queue in Sec. 1.4.2. The first row of the table is a snapshot of the system just after initialization at time 0, the second row is a snapshot of the system just after the first event (an arrival) has occurred, etc.

Technique 5

The model should be run, when possible, under simplifying assumptions for which its true characteristics are known or can easily be computed.

EXAMPLE 5.10. For the job-shop model presented in Sec. 2.7, it is not possible to compute the desired system characteristics analytically. Therefore, one must use simulation. To debug the simulation model, one could first run the general model of Sec. 2.7.2 with one workstation, one machine in that station, and only type 1 jobs (which have an arrival rate of $0.3/0.25 = 1.2$ jobs per hour). The resulting model is known as the $M/E_2/1$ queue and has known transient and steady-state characteristics [see Kelton (1985) and Gross and Harris (1998, p. 132)]. Table 5.2 gives the theoretical values of the steady-state average number in queue, average utilization, and average delay in queue, and also estimates of these quantities from a simulation run of length 2000 eight-hour days. Since the estimates are very close to the true values, we have some degree of confidence that the computer program is correct.

A more definitive test of the program can be achieved by running the general model of Sec. 2.7.2 with the original number of workstations (5), the original number of machines in each station (3, 2, 4, 3, 1), only type 1 jobs, and with exponential service times (with the same mean as the corresponding 2-Erlang service time) at each workstation. The resulting model is, in effect, four multiserver queues in series, with the first queue an $M/M/3$, the second an $M/M/2$, etc. [The interdeparture times from an $M/M/s$ queue (s is the number of servers) that has been in operation for a long time are IID exponential random variables; see Gross and Harris (1998, p. 167).] Furthermore, steady-state characteristics are known for the $M/M/s$ queue [see Gross and Harris (1998, p. 69)]. Table 5.3 gives, for each workstation, the theoretical values of the steady-state average number in queue, average utilization, and average delay in queue, and also estimates of these quantities from a simulation run of length 2000 eight-hour days. Once again, the simulation estimates are quite close to the theoretical values, which gives increased confidence in the program.

EXAMPLE 5.11. We developed a simulation model for a large provider of cellular phone service, where the goal was to determine the long-term availability (proportion of time up) of several alternative network configurations. Originally, we tried computing availability using analytic approaches such as continuous-time Markov chains and

TABLE 5.1
Partial trace for the single-server queue considered in Sec. 1.4.2

Event	Clock	Server status	Number in queue	Times of arrival	Event list: Arrive	Event list: Depart	Number of customers delayed	Total delay	Area under number-in-queue function	Area under busy function
Initialization	0	0	0		0.4	∞	0	0	0	0
Arrival	0.4	1	0		1.6	2.4	1	0	0	0
Arrival	1.6	1	1	1.6	2.1	2.4	1	0	0	1.2
Arrival	2.1	1	2	1.6, 2.1	3.8	2.4	1	0	0.5	1.7
Departure	2.4	1	1	2.1	3.8	3.1	2	0.8	1.1	2.0
Departure	3.1	1	0		3.8	3.3	3	1.8	1.8	2.7
Departure	3.3	0	0		3.8	∞	3	1.8	1.8	2.9

TABLE 5.2
Theoretical values (T) and simulation estimates (S) for a simplified job-shop model ($M/E_2/1$ queue)

Average number in queue		Average utilization		Average delay in queue	
T	S	T	S	T	S
0.676	0.685	0.600	0.604	0.563	0.565

TABLE 5.3
Theoretical values (T) and simulation estimates (S) for a simplified job-shop model (four multiserver queues in series)

Work station	Average number in queue		Average utilization		Average delay in queue	
	T	S	T	S	T	S
3	0.001	0.001	0.150	0.149	0.001	0.001
1	0.012	0.012	0.240	0.238	0.010	0.010
2	0.359	0.350	0.510	0.508	0.299	0.292
5	0.900	0.902	0.600	0.601	0.750	0.752

conditional expectation [see, for example, Ross (2003)], but we were only able to obtain results for simple cases. Therefore, we needed to use simulation, and we partially verified our simulation model by comparing the simulation and analytic results for the simple cases.

Technique 6

With some types of simulation models, it may be helpful to observe an animation of the simulation output (see Sec. 3.4.3).

EXAMPLE 5.12. A simulation model of a network of automobile traffic intersections was developed, supposedly debugged, and used for some time to study such issues as the effect of various light-sequencing policies. However, when the simulated flow of traffic was animated, it was found that simulated cars were actually colliding in the intersections; subsequent inspection of the computer program revealed several previously undetected errors.

Technique 7

Compute the sample mean and sample variance for each simulation input probability distribution, and compare them with the desired (e.g., historical) mean and variance. This suggests that values are being correctly generated from these distributions.

EXAMPLE 5.13. The parameters of gamma and Weibull distributions are defined differently in various simulation packages and books. Thus, this technique would be valuable here.

Technique 8

Use a commercial simulation package to reduce the amount of programming required. On the other hand, care must be taken when using a simulation package (particularly a recently released one), since it may contain errors of a subtle nature. Also, simulation packages contain powerful high-level macro statements, which are sometimes not well documented.

5.4 TECHNIQUES FOR INCREASING MODEL VALIDITY AND CREDIBILITY

In this section we discuss six classes of techniques for increasing the validity and credibility of a simulation model.

5.4.1 Collect High-Quality Information and Data on the System

In developing a simulation model, the analyst should make use of all existing information, including the following:

Conversations with Subject-Matter Experts

A simulation model is not an abstraction developed by an analyst working in isolation; in fact, the modeler must work closely with people who are intimately familiar with the system. There will never be one single person or document that contains all the information needed to build the model. Therefore, the analyst will have to be resourceful to obtain a complete and accurate set of information. Care must be taken to identify the true SMEs for each subsystem and to avoid obtaining biased data (see Example 5.19). The process of bringing all the system information together in one place is often valuable in its own right, even if a simulation study is never performed. Note that since the specifications for a system may be changing during the course of a simulation study, the modeler may have to talk to some SMEs on a continuing basis.

EXAMPLE 5.14. For a manufacturing system, the modelers should obtain information from sources such as machine operators, manufacturing and industrial engineers, maintenance personnel, schedulers, managers, vendors, and blueprints.

EXAMPLE 5.15. For a communications network, relevant people might include end-users, network designers, technology experts (e.g., for switches and satellites), system administrators, application architects, maintenance personnel, managers, and carriers.

Observations of the System

If a system similar to the one of interest exists, then data should be obtained from it for use in building the model. These data may be available from historical records or may have to be collected during a time study. Since the people who provide the data might be different from the simulation modelers, it is important that the following two principles be followed:

- The modelers need to make sure that the data requirements (type, format, amount, conditions under which they should be collected, why needed, etc.) are specified precisely to the people who provide the data.
- The modelers need to understand the process that produced the data, rather than treat the observations as just abstract numbers.

The following are five potential difficulties with data:

- Data are not representative of what one really wants to model.

EXAMPLE 5.16. The data that have been collected during a military field test may not be representative of actual combat conditions due to differences in troop behavior and lack of battlefield smoke (see also Prob. 5.1).

- Data are not of the appropriate type or format.

EXAMPLE 5.17. In modeling a manufacturing system, the largest source of randomness is usually random downtimes of a machine. Ideally, we would like data on time to failure (in terms of actual machine busy time) and time to repair of a machine. Sometimes data are available on machine breakdowns, but quite often they are not in the proper format. For example, the times to failure might be based on wall-clock time and include periods that the machine was idle or off-shift.

- Data may contain measurement, recording, or rounding errors.

EXAMPLE 5.18. Repair times for military-aircraft components were often rounded to the nearest day, making it impossible to fit a continuous probability distribution (see Chap. 6).

- Data may be "biased" because of self-interest.

EXAMPLE 5.19. The maintenance department in an automotive factory reported the reliability of certain machines to be greater than reality to make themselves look good.

- Data may have inconsistent units.

EXAMPLE 5.20. The U.S. Transportation Command transports military cargo by air, land, and sea. Sometimes there is confusion in building simulation models because the U.S. Air Force and the U.S. Army use short tons (2000 pounds) while the U.S. Navy uses long tons (2200 pounds).

Existing Theory

For example, if one is modeling a service system such as a bank and the arrival rate of customers is constant over some time period, theory tells us that the interarrival times of customers are quite likely to be IID exponential random variables; in other words, customers arrive in accordance with a Poisson process (see Sec. 6.12.1 and Example 6.4).

Relevant Results from Similar Simulation Studies

If one is building a simulation model of a military ground encounter (as has been done many times in the past), then results from similar studies should be sought out and used, if possible.

Experience and Intuition of the Modelers

It will often be necessary to use one's experience or intuition to hypothesize how certain components of a complex system operate, particularly if the system does not currently exist in some form. It is hoped that these hypotheses can be substantiated later in the simulation study.

5.4.2 Interact with the Manager on a Regular Basis

We now discuss one of the most important ideas in this chapter, whose use will increase considerably the likelihood that the completed model will be employed in the decision-making process. *It is extremely important for the modeler to interact with the manager on a regular basis throughout the course of the simulation study.* This approach has the following benefits:

- When a simulation study is initiated, there may not be a clear idea of the problem to be solved. Thus, as the study proceeds and the nature of the problem becomes clearer, this information should be conveyed to the manager, who may reformulate the study's objectives. Clearly, the greatest model for the wrong problem is invalid!
- The manager's interest and involvement in the study are maintained.
- The manager's knowledge of the system contributes to the actual validity of the model.
- The model is more credible since the manager understands and accepts the model's assumptions. As a matter of fact, it is extremely desirable to have the manager (and other important personnel) "sign off" on key model assumptions. This may cause the manager to believe, "Of course, it's a good model, since I helped develop it."

5.4.3 Maintain a Written Assumptions Document and Perform a Structured Walk-Through

Communication errors are a major reason why simulation models often contain invalid assumptions or have critical omissions. The documentation of all model concepts, assumptions, algorithms, and data summaries in a written *assumptions document* can greatly lessen this problem, and it will also enhance the credibility of the model. (Within DoD an assumptions document is better known as a *conceptual model.*) However, deciding on the appropriate content of an assumptions document is a less-than-obvious task that depends on the modeler's insight, knowledge of modeling principles (e.g., from operations research, probability and statistics, etc.), and experience in modeling similar types of systems. An assumptions document is not an "exact" description of how the system works, but rather a description of how it works relative to the particular issues that the model is to address. Indeed, the assumptions document is the embodiment of the simulation analyst's vision of how the system of interest should be modeled.

The assumptions document should be written to be readable by analysts, SMEs, and technically trained managers alike, and it should contain the following:

- An overview section that discusses overall project goals, the specific issues to be addressed by the simulation study, and the performance measures for evaluation.
- A process-flow or system-layout diagram, if appropriate.
- Detailed descriptions of each subsystem *in bullet format* and how these subsystems interact. (Bullet format, as on this page, makes the assumptions document easier to review at the structured walk-through of the assumptions document, which is described below.)
- What simplifying assumptions were made and why. Remember that a simulation model is supposed to be a simplification or abstraction of reality.
- Limitations of the simulation model.
- Summaries of a data set such as its sample mean and a histogram. Detailed statistical analyses or other technical material should probably be placed in appendices to the report—remember that the assumptions document should be readable by technical managers.
- Sources of important or controversial information.

The assumptions document should contain enough detail that it is a "blueprint" for creating the simulation computer program. Additional information on assumptions documents (conceptual models) can be found in Defense Modeling and Simulation Office (2000), Pace (2003), and Robinson (2004).

As previously discussed, the simulation modeler will need to collect system information from many different people. Furthermore, these people are typically very busy dealing with the daily problems that occur within their organizations, often resulting in their giving something less than their undivided attention to the questions posed by the simulation modeler. As a result, there is a considerable danger that the simulation modeler will not obtain a complete and correct description of the system. *One way of dealing with this potential problem is to conduct a structured walk-through of the assumptions document before an audience of SMEs and managers.* Using a projection device, the simulation modeler goes through the assumptions document bullet by bullet, but not proceeding from one bullet to the next until everybody in the room is convinced that a particular bullet is correct and at an appropriate level of detail. A structured walk-through will increase both the validity and the credibility of the simulation model.

The structured walk-through ideally should be held at a remote site (e.g., a hotel meeting room), so that people give the meeting their full attention. Furthermore, it should be held prior to the beginning of programming in case major problems are uncovered at the meeting. The assumptions document should be sent to participants prior to the meeting and their comments requested. We do not, however, consider this to be a replacement for the structured walk-through itself, since people may not have the time or motivation to review the document carefully on their own. Furthermore, the interactions that take place at the actual meeting are invaluable. [Within DoD the structured walk-through of the assumptions document (conceptual model) is sometimes called *conceptual model validation*.] It is imperative that all key members of the project team be present at the structured walk-through and that they all take an active role.

It is likely that *many* model assumptions will be found to be incorrect or to be missing at the structured walk-through. Thus, any errors or omissions found in the assumptions document should be corrected before programming begins.

We now present two examples of structured walk-throughs, the first being very successful and the other producing quite surprising but still useful results.

EXAMPLE 5.21. We performed a structured walk-through in doing a simulation study for a Fortune 500 manufacturing company (see Sec. 13.6). There were nine people at the meeting, including two modelers and seven people from the client organization. The client personnel included the foreman of the machine operators, three engineers of various types, two people from the scheduling department, and a manager. The assumptions document was 19 pages long and contained approximately 160 tentative model assumptions. Each of the 160 assumptions was presented and discussed, with the whole process taking 5½ hours. The process resulted in several erroneous assumptions being discovered and corrected, a few new assumptions being added, and some level-of-detail issues being resolved. Furthermore, at the end of the meeting, all nine people felt that *they* had a valid model! In other words, they had taken *ownership* of the model.

EXAMPLE 5.22. At a structured walk-through for a transportation system, a significant percentage of the assumptions given to us by our corporate sponsor were found to be wrong by the SMEs present. (Due to the long geographic distances between the home offices of the sponsor and the SMEs, it was not possible for the SMEs to be present at the kickoff meeting for the project.) As a result, various people were assigned responsibilities to collect information on different parts of the system. The collected information was used to update the assumptions document, and a second walk-through was successfully performed. This experience pointed out the critical importance of having all key project members present at the kickoff meeting.

5.4.4 Validate Components of the Model by Using Quantitative Techniques

The simulation analyst should use quantitative techniques whenever possible to test the validity of various components of the overall model. We now give some examples of techniques that can be used for this purpose, all of which are generally applicable.

If one has fitted a theoretical probability distribution to a set of observed data, then the adequacy of the representation can be assessed by using the graphical plots and goodness-of-fit tests discussed in Chap. 6.

As stated in Sec. 5.4.1, it is important to use appropriate data in building a model; however, it is equally important to exercise care when structuring these data. For example, if several sets of data have been observed for the "same" random phenomenon, then the correctness of merging these data can be assessed by the Kruskal-Wallis test of homogeneity of populations (see Sec. 6.13). If the data sets appear to be homogeneous, they can be merged and the combined data set used for some purpose in the simulation model.

EXAMPLE 5.23. For the manufacturing system described in the case study of Sec. 13.6, time-to-failure and time-to-repair data were collected for two "identical" machines made by the same vendor. However, the Kruskal-Wallis test showed that the two distributions were, in fact, different for the two machines. Thus, each machine was given its own time-to-failure and time-to-repair distributions in the simulation model.

An important technique for determining which model factors have a significant impact on the desired measures of performance is *sensitivity analysis*. If a particular factor appears to be important, then it needs to be modeled carefully. The following are examples of factors that could be investigated by a sensitivity analysis:

- The value of a parameter (see Example 5.24)
- The choice of a distribution
- The entity moving through the simulated system (see Example 5.25)
- The level of detail for a subsystem (see Example 5.26)
- What data are the most crucial to collect (using a "coarse" model of the system)

EXAMPLE 5.24. In a simulation study of a new system, suppose that the value of a parameter is estimated to be 0.75 as a result of conversations with SMEs. The importance of this parameter can be determined by running the simulation with 0.75 and, in addition, by running it with each of the values 0.70 and 0.80. If the three simulation runs produce approximately the same results, then the output is not sensitive to the choice of the parameter *over the range 0.70 to 0.80*. Otherwise, a better specification of the parameter is needed.

EXAMPLE 5.25. We built a simulation model for a candy-bar manufacturing line. Initially, we used a single candy bar as the basic entity moving through the model, but this resulted in excessive computer execution time. A sensitivity analysis was performed, and it was found that using one-quarter of a case of candy bars (150 candy bars) produced virtually the same simulation results for the desired performance measure, *cases produced per shift*, while reducing the execution time considerably.

EXAMPLE 5.26. We developed a simulation model of the assembly and test area for a PC manufacturing company. Later the company managers decided that they wanted to run the model on their own computers, but the memory requirements of the model were too great. As a result, we were forced to simplify greatly the model of the assembly area to save computer memory. (The main focus of the simulation study was the required capacity for the test area.) We ran the simplified simulation model (the model of the test area was unchanged) and found that the desired performance measure, *daily throughput*, differed by only 2 percent from that of the original model. Thus, a large amount of detail was unnecessary for the assembly area. Note, however, that the simplified model would not have been appropriate to study how to improve the efficiency of the assembly area. On the other hand, it may not have been necessary to model the test area in this case.

When one is performing a sensitivity analysis, it is important to use the method of common random numbers (see Sec. 11.2) to control the randomness in the simulation. Otherwise, the effect of changing one factor may be confounded with other changes (e.g., different random values from some input distribution) that inadvertently occur.

If one is trying to determine the sensitivity of the simulation output to changes in two or more factors of interest, then it is not correct, in general, to vary one factor at a time while setting the other factors to some arbitrary values. A more correct approach is to use statistical experimental design, which is discussed in Chap. 12. The effect of each factor can be formally estimated; and if the number of factors is not too large, interactions between factors can also be detected.

5.4.5 Validate the Output from the Overall Simulation Model

The most definitive test of a simulation model's validity is to establish that its output data closely resemble the output data that would be expected from the actual (proposed) system. This might be called *results validation* and, in this section, we will discuss several ways that it could be carried out.

Comparison with an Existing System

If a system similar to the proposed one now exists, then a simulation model of the existing system is developed and its output data are compared to those from the existing system itself. If the two sets of data compare "closely," then the model of the *existing* system is considered "valid." (The accuracy required from the model will depend on its intended use and the utility function of the manager.) The model is then modified so that it represents the proposed system. The greater the commonality between the existing and proposed systems, the greater our confidence in the model of the proposed system. There is no completely definitive approach for validating the model of the proposed system. If there were, there might be no need for a simulation model in the first place. If the above comparison is successful, then it has the additional benefit of providing credibility for the use of simulation (see Example 5.27). The comparison of the model and system output data could be done using numerical statistics such as the sample mean, the sample variance, and the sample correlation function. Alternatively, the assessment could be made by using graphical plots (see Example 5.30) such as histograms, distribution functions, box plots, and spider-web plots (called radar plots in Microsoft Excel).

EXAMPLE 5.27. We performed a simulation study for the corporate headquarters of a manufacturer of paper products. A particular manufacturing plant for this company currently had two machines of a certain type, and local management wanted to purchase a third machine. The goal of the study was to see whether the additional machine was really needed. To validate our model, we first simulated the existing system with two machines. The model and system throughputs for the two machines differed by 0.4 and 1.1 percent, while the machine utilizations differed by 1.7 and 11 percent. (The relatively large error of 11 percent was caused by the second machine operator's not following company policy.) Using the "validated" simulation model, we simulated the system with three machines and found that the additional machine was not necessary. Based on the *credible* simulation results, the vice president for manufacturing of the entire company rejected the plant's request for a new machine, resulting in a capital avoidance of $1.4 million.

EXAMPLE 5.28. A U.S. Air Force test agency performed a simulation study for a wing of bombers using the Logistics Composite Model (LCOM). The ultimate goal of the study was to evaluate the effect of various proposed logistics policies on the availability of the bombers, i.e., the proportion of time that the bombers were available to fly missions. Data were available from the actual operations of the wing over a 9-month period, and they included both failure data for various aircraft components and the wing availability. To validate the model, the Air Force first simulated the 9-month period with the existing logistics policy. The model availability differed from the historical availability by less than 3 percent, providing strong evidence for the validity of the model.

EXAMPLE 5.29. A major manufacturer of telecommunications switches submitted a prototype switch to an artificial traffic stream (e.g., exponential interarrival times) in a laboratory. A simulation model of the switch was then submitted to the same traffic stream, and comparable model and system performance measures were compared. The closeness of the respective measures gave the model developers confidence in the validity of the model.

EXAMPLE 5.30. A hypothetical new ground-to-ground missile is being developed by the U.S. Army. Eight prototype missiles were field tested for the *same* scenario (and set of environmental conditions), and their impact points in an xy coordinate system were recorded. A simulation model for the missile system was developed, 15 independent replications of the model were made for the same scenario using different random numbers, and the corresponding impact points were computed. The impact points for the test and simulated missiles (in feet) are plotted in Fig. 5.2. It appears from the figure that the simulated missiles are less accurate than the test missiles, but it would be desirable to have further substantiation. We next computed the miss distance d for each test missile and each simulated missile using the Pythagorean theorem, which states that

$$d = \sqrt{x^2 + y^2}$$

The resulting miss distances (in feet) are given in Table 5.4, where it's seen that the average miss distance for the simulated missiles is 14.7 percent larger than the average miss distance for the test missiles. A spider-web plot for the miss distances is given in Fig. 5.3.

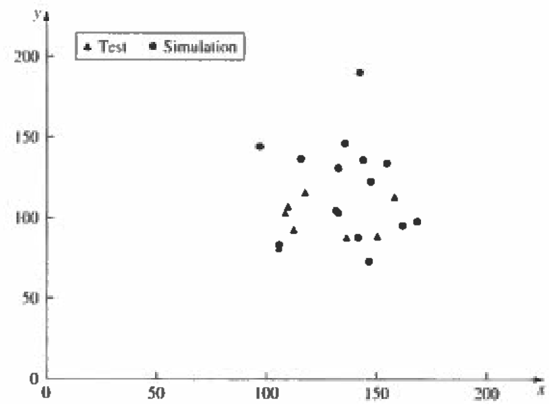

FIGURE 5.2
Impact points for the test and simulated missiles (in feet).

TABLE 5.4
Miss distances d for the test and simulated missiles (in feet)

Missile number	Test miss distance	Simulation miss distance
1	174.45	134.60
2	146.09	194.73
3	194.72	168.14
4	149.84	178.82
5	161.93	163.78
6	165.52	186.39
7	153.62	237.20
8	133.46	187.73
9	—	197.90
10	—	173.55
11	—	166.64
12	—	199.10
13	—	168.17
14	—	204.32
15	—	191.48
Sample mean	159.95	183.50
Sample variance	355.75	545.71

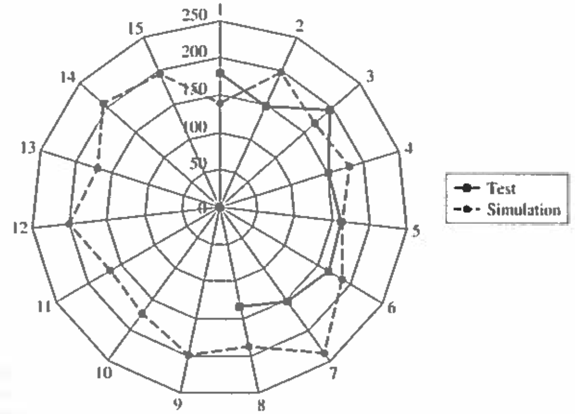

FIGURE 5.3
Spider-web plot for the missile and simulation miss distances.

The numbers 50, 100, . . . , 250 are the miss distances, and the numbers 1, 2, . . . , 15 are the missile numbers. It is clear from this plot that the simulation miss distances are, in general, larger than the test miss distances. In summary, based on the sample means and the two plots, it appears that the model does not provide a valid representation for the prototype missile relative to the criterion of miss distance. However, we will revisit this example in Sec. 5.6.2.

In addition to statistical procedures, one can use a *Turing test* [see Turing (1950), Schruben (1980), and Carson (1986)] to compare the output data from the model to those from the system. People knowledgeable about the system (e.g., engineers or managers) are asked to examine one or more sets of system data as well as one or more sets of model data, without knowing which sets are which. Each data set should be presented on a separate piece of paper using exactly the same format. If these SMEs can differentiate between the system and model data, their explanation of how they were able to do so is used to improve the model.

EXAMPLE 5.31. Schruben (1980) reports the use of a Turing test in a simulation study of an automobile-component factory. Data from the factory and from the simulation were put on time-study forms and reviewed at a meeting by three managers, three industrial engineers, and two factory workers. The inability of these people to agree on which data were real and which were simulated led to immediate acceptance of the simulation model.

EXAMPLE 5.32. An animation version of the Turing test was used in validating a simulation model of microscopic vehicle flow on a freeway. An animation of traffic flow from the simulation was displayed simultaneously on a large-screen monitor with an animation produced from data collected from the actual freeway. The data from the freeway were collected by a video camera mounted on an airplane.

Up to now, we have discussed validating a simulation model relative to past or present system output data; however, a perhaps more definitive test of a model is to establish its ability to predict *future* system behavior. Since models often evolve over time and are used for multiple applications (particularly legacy models within the DoD), there is often an opportunity for such *prospective* validation. For example, if a model is used to decide which version of a proposed system to build, then after the system has been built and sufficient time has elapsed for output data to be collected, these data can be compared with the predictions of the model. If there is reasonable agreement, we have increased confidence in the "validity" of the model. On the other hand, discrepancies between the two data sets should be used to update the model. Regardless of the accuracy of a model's past predictions, a model should be carefully scrutinized before each new application, since a change in purpose or the passage of time may have invalidated some aspect of the existing model. This once again points out the need for good documentation of the model.

Suppose that we compare the output data from an existing system with those from a simulation model of that system and find significant discrepancies. If these discrepancies or other information *objectively* suggests how to improve the model, then these changes should be made and the simulation rerun. If the new simulation output data compare closely with the system output data, then the model can be considered "valid."

Suppose instead that there are major discrepancies between the system and model output data, but that changes are made to the model, somewhat without justification (e.g., some parameter is "tweaked"), and the resulting output data are again compared with the system output data. This procedure, which we call *calibration* of a model, is continued until the two data sets agree closely. However, we must ask whether this procedure produces a valid model for the system, in general, or whether the model is only representative of this particular set of input data. To answer this question (in effect, to validate the model), one can use a completely independent set of system input and output data. The calibrated model might be driven by the second set of input data (in a manner similar to that described in Sec. 5.6.1) and the resulting model output data compared with the second set of system output data. This idea of using one set of data for calibration and another independent set for validation is fairly common in economics and the biological sciences. In particular, it was used by the Crown Zellerbach Corporation in developing a simulation model of tree growth. Here the system data were available from the U.S. Forest Service.

Comparison with Expert Opinion

Whether or not there is an existing system, SMEs should review the simulation results for reasonableness. (Care must be taken in performing this exercise, since if one knew exactly what output to expect, there would be no need for a model.) If the simulation results are consistent with perceived system behavior, then the model is said to have *face validity*.

EXAMPLE 5.33. The above idea was put to good use in the development of a simulation model of the U.S. Air Force manpower and personnel system. (This model was designed to provide Air Force policy analysts with a systemwide view of the effects of various proposed personnel policies.) The model was run under the baseline personnel policy, and the results were shown to Air Force analysts and decision makers, who subsequently identified some discrepancies between the model and perceived system behavior. This information was used to improve the model, and after several additional evaluations and improvements, a model was obtained that appeared to approximate current Air Force policy closely. This exercise improved not only the validity of the model, but also its credibility.

Comparison with Another Model

Suppose that another model was developed for the same system and for a "similar" purpose, and that it is thought to be a "valid" representation. Then numerical statistics or graphical plots for the model that is currently of interest can be informally compared with the comparable statistics or graphical plots from the other model. Alternatively, the confidence-interval procedures discussed in Sec. 10.2 can be used to make a more formal comparison between the two models. It should be kept in mind that just because two models produce similar results doesn't necessarily mean that either model is valid, since both models could contain a similar error.

EXAMPLE 5.34. A defense supply center was building a new simulation model called the Performance and Requirements Impact Simulation to replace an existing model. One of the purposes of both models is to decide when to order and how much to order for each stock number. To validate the *old model*, the total dollar amount of all orders placed by the model for fiscal year 1996 was compared with the total dollar amount for the actual system for the same time period. Since these dollar amounts differed by less than 3 percent, there was a fair amount of confidence in the validity of the old model. To validate the *new model*, the two models were used to predict the total dollar amount of all orders for fiscal year 1998, and the results differed by less than 6 percent. Thus, there was reasonable confidence in the validity of the new model.

In Example 5.34, it probably would have been a good idea for the simulation analysts to also use a smaller level of aggregation for validation purposes, such as the dollar amounts for certain categories of stock numbers. (It is possible that positive errors for some categories might cancel out negative errors for other categories.) Also, it would have been interesting to compare the total dollar amounts for all orders placed by the two models in 1996.

5.4.6 Animation

An animation can be an effective way to find invalid model assumptions and to enhance the credibility of a simulation model.

EXAMPLE 5.35. A simulation model was developed for a candy packaging system. A newly promoted operations manager, who had no familiarity with the simulation model, declared, "That's my system!" upon seeing an animation of his system for the first time—the model gained instant credibility.

5.5 MANAGEMENT'S ROLE IN THE SIMULATION PROCESS

The manager of the system of interest must have a basic understanding of simulation and be aware that a successful simulation study requires a commitment of his or her *time* and resources. The following are some of the responsibilities of the manager:

- Formulating problem objectives
- Directing personnel to provide information and data to the simulation modeler and to attend the structured walk-through
- Interacting with the simulation modeler on a regular basis
- Using the simulation results as an aid in the decision-making process

Simulation studies require the use of an organization's technical personnel for some period of time. If the study is done in-house, then several company personnel may be required full-time for several months. These people often have other jobs such as being responsible for the day-to-day operations of a manufacturing system. Even if a consultant does the study, company personnel must be involved in the modeling process and may also be needed to collect data.

5.6 STATISTICAL PROCEDURES FOR COMPARING REAL-WORLD OBSERVATIONS AND SIMULATION OUTPUT DATA

In this section we present statistical procedures that might be useful for carrying out the comparison of model and system output data (see Sec. 5.4.5).

Suppose that $R_1, R_2, \ldots, R_k$ are observations from a real-world system and that $M_1, M_2, \ldots, M_l$ are output data from a corresponding simulation model (see Example 5.36). We would like to compare the two data sets in some way to determine whether the model is an accurate representation of the real-world system. The first approach that comes to mind is to use one of the classical statistical tests (t, Mann-Whitney, two-sample chi-square, two-sample Kolmogorov-Smirnov, etc.) to determine whether the underlying distributions of the two data sets can be safely regarded as being the same. [For a good discussion of these tests, which assume *IID data*, see Breiman (1973) and Conover (1999).] However, the output processes of almost all real-world systems and simulations are *nonstationary* (the distributions of the successive observations change over time) and *autocorrelated* (the observations in the process are correlated with each other), and thus none of these tests is *directly* applicable. Furthermore, we question whether hypothesis tests, as compared with constructing confidence intervals for differences, are even the appropriate statistical approach. Since the model is only an approximation to the actual system, a null hypothesis that the system and model are the "same" is clearly false. We believe that it is more useful to ask whether the differences between the system and the model are significant enough to affect any conclusions derived from the model. In Secs. 5.6.1 through 5.6.3 we discuss, respectively, inspection, confidence-interval, and time-series approaches to this comparison problem. Finally, two additional approaches based on regression analysis and bootstrapping are discussed in Sec. 5.6.4.

5.6.1 Inspection Approach

The approach that seems to be used by most simulation practitioners who attempt the aforementioned comparison is to compute one or more numerical statistics from the real-world observations and corresponding statistics from the model output data, and then compare the two sets of statistics without the use of a formal statistical procedure (see Examples 5.27 and 5.28). Examples of statistics that might be used for this purpose are the sample mean, the sample variance (see Sec. 4.4 for a discussion of the danger in using the sample variance from autocorrelated data), and the sample correlation function. (The comparison of graphical plots can also be quite useful, as we saw in Example 5.30.) The difficulty with this inspection approach, which is graphically illustrated on the next page in Example 5.36, is that each statistic is essentially a sample of size 1 from some underlying population, making this idea particularly vulnerable to the inherent randomness of the observations from both the real system and the simulation model.