10

Verification and Validation of Simulation Models

One of the most important and difficult tasks facing a model developer is the verification and validation of the simulation model. The engineers and analysts who use the model outputs to aid in making design recommendations and the managers who make decisions based on these recommendations justifiably look upon a model with some degree of skepticism about its validity. It is the job of the model developer to work closely with the end users throughout the period of development and validation to reduce this skepticism and to increase the model's credibility. The goal of the validation process is twofold: (1) to produce a model that represents true system behavior closely enough for the model to be used as a substitute for the actual system for the purpose of experimenting with the system; and (2) to increase to an acceptable level the credibility of the model, so that the model will be used by managers and other decision makers.

Validation should not be seen as an isolated set of procedures that follows model development, but rather as an integral part of model development. Conceptually, however, the verification and validation process consists of the following components:

1. Verification is concerned with building the model right. It is utilized in the comparison of the conceptual model to the computer representation that implements that conception. It asks the questions: Is the model implemented correctly in the computer? Are the input parameters and logical structure of the model correctly represented?
2. Validation is concerned with building the right model. It is utilized to determine that a model is an accurate representation of the real system. Validation is usually achieved through the calibration of the model, an

iterative process of comparing the model to actual system behavior and using the discrepancies between the two, and the insights gained, to improve the model. This process is repeated until model accuracy is judged to be acceptable.

This chapter describes methods that have been recommended and used in the verification and validation process. Most of the methods are informal subjective comparisons, while a few are formal statistical procedures. The use of the latter procedures involves issues related to output analysis, the subject of Chapters 11 and 12. Output analysis refers to analyzing the data produced by a simulation and drawing inferences from these data about the behavior of the real system. To summarize their relationship, validation is the process by which model users gain confidence that output analysis is making valid inferences about the real system under study.

Many articles and chapters in texts have been written on verification and validation. For discussion of the main issues, the reader is referred to Balci [1994, 1998], Carson [1986], Gass [1983], Kleijnen [1995], Law and Kelton [2000], Naylor and Finger [1967], Oren [1981], Sargent [1994], Shannon [1975], and van Horn [1969, 1971]. For statistical techniques relevant to different aspects of validation, the reader can obtain the foregoing references plus those by Balci and Sargent [1982a, b; 1984a], Kleijnen [1987], and Schruben [1980]. For case studies in which validation is emphasized, the reader is referred to Carson et al. [1981a,b], Gafarian and Walsh [1970], Kleijnen [1993], and Shechter and Lucas [1980]. Bibliographies on validation have been published by Balci and Sargent [1984b] and Youngblood [1993].

10.1 Model Building, Verification, and Validation

The first step in model building consists of observing the real system and the interactions among its various components and collecting data on its behavior. But observation alone seldom yields sufficient understanding of system behavior. Persons familiar with the system, or any subsystem, should be questioned to take advantage of their special knowledge. Operators, technicians, repair and maintenance personnel, engineers, supervisors, and managers understand certain aspects of the system which may be unfamiliar to others. As model development proceeds, new questions may arise, and the model developers will return to this step of learning true system structure and behavior.

The second step in model building is the construction of a conceptual model—a collection of assumptions on the components and the structure of the system, plus hypotheses on the values of model input parameters. As illustrated by Figure 10.1, conceptual validation is the comparison of the real system to the conceptual model.

The third step is the translation of the operational model into a computer-recognizable form — the computerized model. In actuality, model building is not a linear process with three steps. Instead, the model builder will return

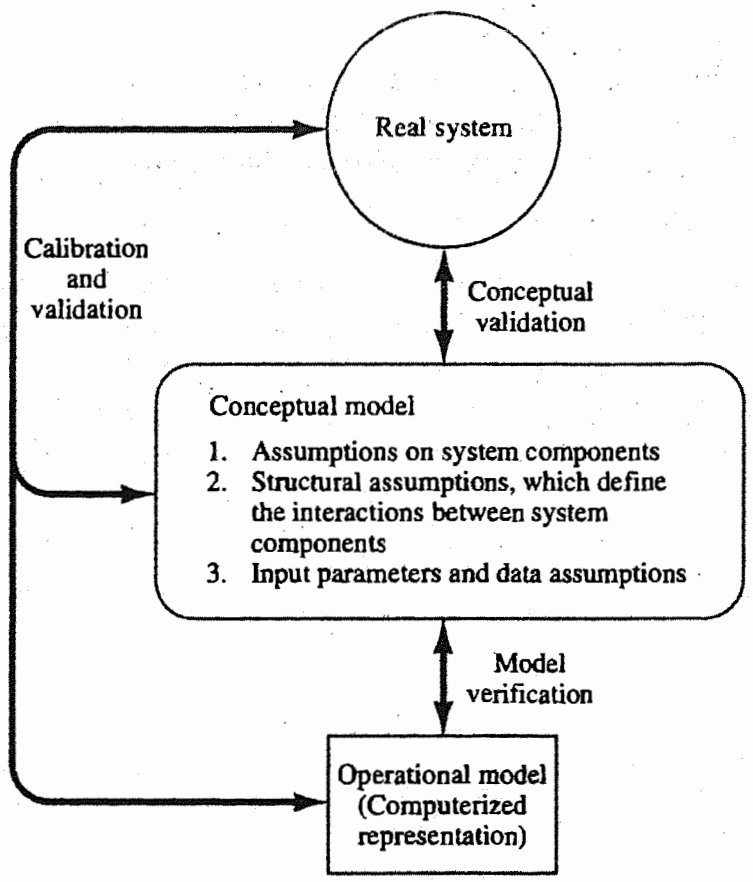

Figure 10.1. Model building, verification, and validation.

to each of these steps many times while building, verifying, and validating the model. Figure 10.1 depicts the ongoing model-building process in which the need for verification and validation causes continual comparison of the real system to the conceptual model and to the operational model, and repeated modification of the model to improve its accuracy.

10.2 Verification of Simulation Models

The purpose of model verification is to assure that the conceptual model is reflected accurately in the computerized representation. The conceptual model quite often involves some degree of abstraction about system operations, or some amount of simplification of actual operations. Verification asks the question: Is the conceptual model (assumptions on system components and system structure, parameter values, abstractions and simplifications) accurately represented by the operational model (i.e., by the computerized representation)?

Many common-sense suggestions can be given for use in the verification process.

1. Have the computerized representation checked by someone other than its developer.
2. Make a flow diagram which includes each logically possible action a system can take when an event occurs, and follow the model logic for each action for each event type. (An example of a logic flow diagram is given in Figures 2.2 and 2.3 for the model of a single-server queue.)

3. Closely examine the model output for reasonableness under a variety of settings of the input parameters. Have the computerized representation print out a wide variety of output statistics.
4. Have the computerized representation print the input parameters at the end of the simulation, to be sure that these parameter values have not been changed inadvertently.
5. Make the computerized representation as self-documenting as possible. Give a precise definition of every variable used and a general description of the purpose of each major section of code.
6. If the computerized representation is animated, verify that what is seen in the animation imitates the actual system. Examples of errors that can be observed through animation are automated guided vehicles (AGVs) that pass through one another at an intersection and entities that disappear (unintentionally) during a simulation.
7. The interactive run controller (IRC) or debugger is an essential component of successful simulation model building. Even the best of simulation analysts makes mistakes or commits logical errors when building a model. The IRC assists in finding and correcting those errors in the following ways:
   (a) The simulation can be monitored as it progresses. This can be accomplished by advancing the simulation until a desired time has elapsed, then displaying model information at that time. Another possibility is to advance the simulation until a particular condition is in effect and then display information.
   (b) Attention can be focused on a particular line of logic or multiple lines of logic that constitute a procedure or a particular entity. For instance, every time that an entity enters a specified block, the simulation will pause so that information can be gathered. As another example, every time that a specified entity becomes active, the simulation will pause.
   (c) Values of selected model components can be observed. When the simulation has paused, the current value or status of variables, attributes, queues, resources, counters, etc., can be observed.
   (d) The simulation can be temporarily suspended, or paused, not only to view information but also to reassign values or redirect entities.
8. Graphical interfaces are recommended for accomplishing verification and validation (Bortscheller and Saulnier [1992]). The graphical representation of the model is essentially a form of self-documentation. It simplifies the task of model understanding.

These suggestions are basically the same ones any software engineer would follow.

Among these common-sense suggestions, one that is most easily implemented, but quite often overlooked, especially by students who are learning simulation, is a close and thorough examination of model output for reasonableness (suggestion 3). For example, consider a model of a complex network of queues consisting of many service centers in series and parallel configurations. Suppose that the modeler is interested mainly in the response time, defined as the time required for a customer to pass through a designated part of the network. During the verification (and calibration) phase of model development, it is recommended that the program collect and print out many statistics in addition to response times, such as utilizations of servers and time-average number of customers in various subsystems. Examination of the utilization of a server, for example, may reveal that it is unreasonably low (or high), a possible error that may be caused by wrong specification of mean service time, or a mistake in model logic that sends too few (or too many) customers to this particular server, or any number of other possible parameter misspecifications or errors in logic.

In a simulation language which automatically collects many standard statistics (average queue lengths, average waiting times, etc.), it takes little or no extra programming effort to display almost all statistics of interest. The effort required can be considerably greater in a general-purpose language such as FORTRAN, C, or C++, which does not have statistics-gathering capabilities to aid the programmer.

Two sets of statistics that can give a quick indication of model reasonableness are *current contents* and *total count*. These statistics apply to any system having items of some kind flowing through it, whether these items are called customers, transactions, inventory, or vehicles. Current contents refers to the number of items in each component of the system at a given time. Total count refers to the total number of items that have entered each component of the system by a given time. In some simulation software, such as GPSS/H and AutoMod, these statistics are automatically kept and can be displayed at any point in simulation time. In other simulation software, simple counters may have to be added to the computerized model and displayed at appropriate times. If the current contents in some portion of the system is high, this indicates that a large number of entities are delayed. If the output is displayed for successively longer simulation run times and the current contents tends to grow in a more or less linear fashion, it is highly likely that a queue is unstable and the server(s) will fall further behind as time continues. This indicates that possibly the number of servers is too small or a service time is misspecified. (Unstable queues were discussed in Chapter 6.) On the other hand, if the total count for some subsystem is zero, this indicates that no items entered that subsystem — again a highly suspect occurrence. Another possibility is that the current count and total count are equal to one. This may indicate that an entity has captured a resource but never freed it. Careful evaluation of these statistics for various run lengths can aid in the detection of mistakes in model logic and data misspecifications. Checking for output reasonableness will usually fail to detect the more

subtle errors, but it is one of the quickest ways to discover gross errors. To aid in error detection, it is best if the model developer forecasts a reasonable range for the value of selected output statistics before making a run of the model. Such a forecast reduces the possibility of rationalizing a discrepancy and failing to investigate the cause of unusual output.

For certain models, it is possible to consider more than whether a particular statistic is reasonable. It is possible to compute certain long-run measures of performance. For example, as seen in Chapter 6, the analyst can compute the long-run server utilization for a large number of queueing systems without any special assumptions regarding interarrival or service-time distributions. Typically, the only information needed is the network configuration plus arrival and service rates. Any measure of performance that can be computed analytically and then compared to its simulated counterpart provides another valuable tool for verification. Presumably, the objective of the simulation is to estimate some measure of performance, such as mean response time, which cannot be computed analytically. But as illustrated by the formulas in Chapter 6 for a number of special queues ($M/M/1$, $M/G/1$, etc.), all the measures of performance in a queueing system are interrelated. Thus, if a simulation model is predicting one measure (such as utilization) correctly, then confidence in the model's predictive ability for other related measures (such as response time) is increased (even though the exact relation between the two measures is, of course, unknown in general and varies from model to model). Conversely, if a model incorrectly predicts utilization, its prediction of other quantities, such as mean response time, is highly suspect.

Another important way to aid the verification process is the oft-neglected documentation phase. If a model builder writes brief comments in the computerized model, plus definitions of all variables and parameters, and descriptions of each major section of the computerized model, it becomes much simpler for someone else, or the model builder at a later date, to verify the model logic. Documentation is also important as a means of clarifying the logic of a model and verifying its completeness.

A more sophisticated technique is the use of a trace. In general, a trace is a detailed computer printout which gives the value of every variable (in a specified set of variables) in a computer program, every time that one of these variables changes in value. A trace designed specifically for use in a simulation program would give the value of selected variables each time the simulation clock was incremented (i.e., each time an event occurred). Thus, a simulation trace is nothing more than a detailed printout of the state of the simulation model as it changes over time.

EXAMPLE 10.1

When verifying the computer implementation (in a general-purpose language such as FORTRAN, Pascal, C, or C++ or in most simulation languages) of the single-server queue model of Example 2.1, an analyst made a run over 16 units of time and observed that the time-average length of the waiting line was

Definition of Variables:
CLOCK = Simulation clock
EVTYP = Event type (start, arrival, departure, or stop)
NCUST = Number of customers in system at time 'CLOCK'
STATUS = Status of server (1–busy, 0–idle)

State of System Just after the Named Event Occurs:

CLOCK = 0	EVTYP = 'Start'	NCUST = 0	STATUS = 0
CLOCK = 3	EVTYP = 'Arrival'	NCUST = 1	STATUS = 0
CLOCK = 5	EVTYP = 'Depart'	NCUST = 0	STATUS = 0
CLOCK = 11	EVTYP = 'Arrival'	NCUST = 1	STATUS = 0
CLOCK = 12	EVTYP = 'Arrival'	NCUST = 2	STATUS = 1
CLOCK = 16	EVTYP = 'Depart'	NCUST = 1	STATUS = 1

.
.
.

Figure 10.2 Simulation of trace of Example 2.1.

$\hat{L}_Q = 0.4375$ customer, which is certainly reasonable for a short run of only 16 time units. Nevertheless, the analyst decided that a more detailed verification would be of value.

The trace in Figure 10.2 gives the hypothetical printout from simulation time CLOCK = 0 to CLOCK = 16 for the simple single-server queue of Example 2.1. This example illustrates how an error can be found with a trace, when no error was apparent from the examination of the summary output statistics (such as $\hat{L}_Q$). Note that at simulation time CLOCK = 3, the number of customers in the system is NCUST = 1, but the server is idle (STATUS = 0). The source of this error could be incorrect logic, or simply not setting the attribute STATUS to a value of 1 (when coding in a general-purpose language or in most simulation languages).

In any case the error must be found and corrected. Note that the less sophisticated practice of examining the summary measures, or output, did not detect the error. By using Equation (6.1), the reader can verify that $\hat{L}_Q$ was computed correctly from the data ($\hat{L}_Q$ is the time-average value of NCUST minus STATUS):

$$\hat{L}_Q = \frac{(0-0)3 + (1-0)2 + (0-0)6 + (1-0)1 + (2-1)4}{3+2+6+1+4}$$

$$= \frac{7}{16} = 0.4375$$

as previously mentioned. Thus, the output measure, $\hat{L}_Q$, had a reasonable value and was computed correctly from the data, but its value was indeed wrong because the attribute STATUS was not assuming correct values. As seen from Figure 10.2, a trace yields information on the actual history of the model which

is more detailed and informative than the summary measures alone.

Most simulation software has a built-in capability to conduct a trace without the programmer having to do any extensive programming. In addition, a 'print' or 'write' statement can be used to implement a tracing capability in a general-purpose language.

As can be easily imagined, a trace over a large span of simulation time can quickly produce an extremely large amount of computer printout, which would be extremely cumbersome to check in detail for correctness. The purpose of the trace is to verify the correctness of the computer program by making detailed paper-and-pencil calculations. To make this practical, a simulation with a trace is usually restricted to a very short period of time. It is desirable, of course, to ensure that each type of event (such as ARRIVAL) occurs at least once, so that its consequences and effect on the model can be checked for accuracy. If an event is especially rare in occurrence, it may be necessary to use artificial data to force it to occur during a simulation of short duration. This is legitimate, as the purpose is to verify that the effect on the system of the rare event is as anticipated.

Some software allows a selective trace. For example, a trace could be set for specific locations in the computerized model. Any time an entity goes through that location or those locations, a message is written. Another example of a selective trace is to set it on a particular entity. Any time that entity becomes active, the trace is on and a message is written. This trace is very useful in following one entity through the entire computerized model. Another example of a selective trace is to set it for the existence of a particular condition. For example, whenever the queue before a certain resource reaches five or more, turn on the trace. This allows running the simulation until something unusual occurs, then examining the behavior from that point forward in time.

Of the three classes of techniques — the common-sense techniques, thorough documentation, and traces — it is recommended that the first two always be carried out. Close examination of model output for reasonableness is especially valuable and informative. A trace can also provide information if it is selective. The generalized trace can be extremely time consuming. ◀

10.3 Calibration and Validation of Models

Verification and validation, although conceptually distinct, usually are conducted simultaneously by the modeler. Validation is the overall process of comparing the model and its behavior to the real system and its behavior. Calibration is the iterative process of comparing the model to the real system, making adjustments (or even major changes) to the model, comparing the revised model to reality, making additional adjustments, comparing again, and so on. Figure 10.3 shows the relationship of model calibration to the overall validation process. The comparison of the model to reality is carried out by a variety of tests—some subjective and others objective. Subjective tests usually

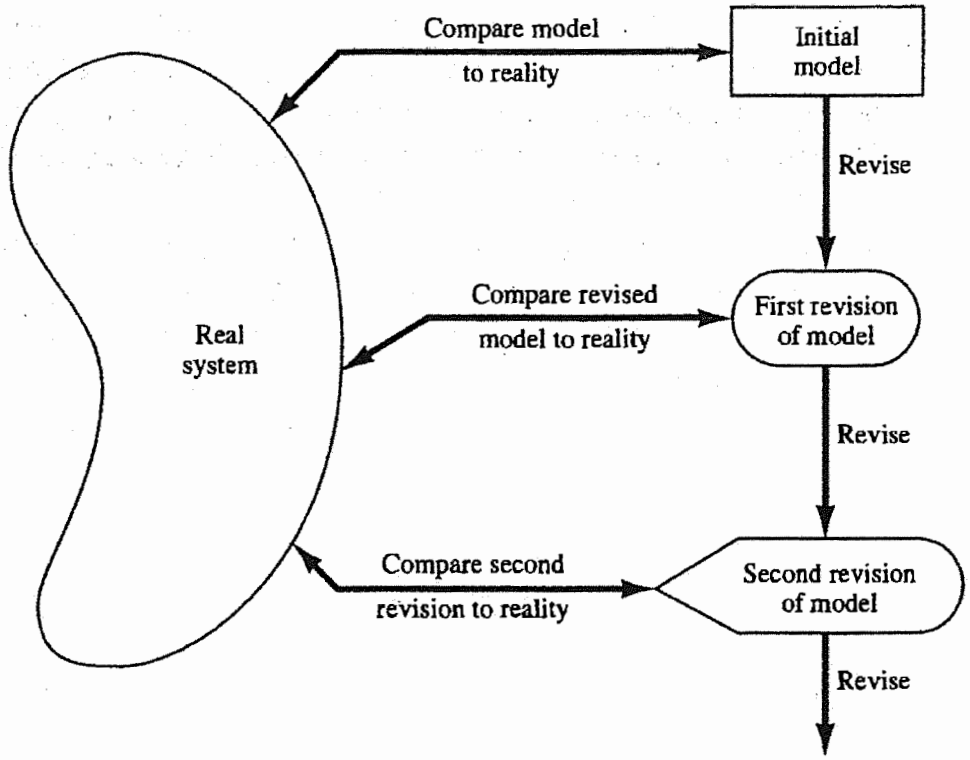

Figure 10.3. Iterative process of calibrating a model.

involve people, who are knowledgeable about one or more aspects of the system, making judgments about the model and its output. Objective tests always require data on the system's behavior plus the corresponding data produced by the model. Then one or more statistical tests are performed to compare some aspect of the system data set to the same aspect of the model data set. This iterative process of comparing model and system, and revising both the conceptual and operational models to accommodate any perceived model deficiencies, is continued until the model is judged to be sufficiently accurate.

A possible criticism of the calibration phase, were it to stop at this point, is that the model has been validated only for the one data set used; that is, the model has been "fit" to one data set. One way to alleviate this criticism is to collect a new set of system data (or to reserve a portion of the original system data) to be used at this final stage of validation. That is, after the model has been calibrated using the original system data set, a "final" validation is conducted using the second system data set. If unacceptable discrepancies between the model and the real system are discovered in the "final" validation effort, the modeler must return to the calibration phase and modify the model until it becomes acceptable.

Validation is not an either/or proposition—no model is ever totally representative of the system under study. In addition, each revision of the model, as pictured in Figure 10.3, involves some cost, time, and effort. The modeler must weigh the possible, but not guaranteed, increase in model accuracy versus the cost of increased validation effort. Usually, the modeler (and model users) have some maximum discrepancy between model predictions and system behavior that would be acceptable. If this level of accuracy cannot be obtained

within the budget constraints, either expectations of model accuracy must be lowered, or the model must be abandoned.

Yücesan and Jacobson [1992] indicate that verifying simulation models is so difficult as to be intractable. They offer theorems to confirm this intractibility.

As an aid in the validation process, Naylor and Finger [1967] formulated a three-step approach which has been widely followed:

1. Build a model that has high face validity.
2. Validate model assumptions.
3. Compare the model input-output transformations to corresponding input-output transformations for the real system.

The next five subsections investigate these three steps in detail.

10.3.1 Face Validity

The first goal of the simulation modeler is to construct a model that appears reasonable on its face to model users and others who are knowledgeable about the real system being simulated. The potential users of a model should be involved in model construction from its conceptualization to its implementation to ensure that a high degree of realism is built into the model through reasonable assumptions regarding system structure, and reliable data. Potential users and knowledgeable persons can also evaluate model output for reasonableness and can aid in identifying model deficiencies. Thus, the users can be involved in the calibration process as the model is iteratively improved, based on the insights gained from the initial model deficiencies. Another advantage of user involvement is the increase in the model's perceived validity, or credibility, without which a manager would not be willing to trust simulation results as a basis for decision making.

Sensitivity analysis can also be used to check a model's face validity. The model user is asked if the model behaves in the expected way when one or more input variables is changed. For example, in most queueing systems, if the arrival rate of customers (or demands for service) were to increase, it would be expected that utilizations of servers, lengths of lines, and delays would tend to increase (although by how much might well be unknown). Based on experience and observations on the real system (or similar related systems), the model user and model builder would probably have some notion at least of the direction of change in model output when an input variable is increased or decreased. For most large-scale simulation models, there are many input variables and thus many possible sensitivity tests. The model builder must attempt to choose the most critical input variables for testing if it is too expensive or time consuming to vary all input variables. If real system data are available for at least two settings of the input parameters, objective scientific sensitivity tests can be conducted using appropriate statistical techniques.

10.3.2 Validation of Model Assumptions

Model assumptions fall into two general classes: structural assumptions and data assumptions. Structural assumptions involve questions of how the system operates and usually involve simplifications and abstractions of reality. For example, consider the customer queueing and service facility in a bank. Customers may form one line, or there may be an individual line for each teller. If there are many lines, customers may be served strictly on a first-come, first-served basis, or some customers may change lines if one is moving faster. The number of tellers may be fixed or variable. These structural assumptions should be verified by actual observation during appropriate time periods together with discussions with managers and tellers regarding bank policies and actual implementation of these policies.

Data assumptions should be based on the collection of reliable data and correct statistical analysis of the data. (Example 9.1 discussed similar issues for a model of a laundromat.) For example, in the bank study previously mentioned, data were collected on:

1. Interarrival times of customers during several 2-hour periods of peak loading ("rush-hour" traffic)
2. Interarrival times during a slack period
3. Service times for commercial accounts
4. Service times for personal accounts

The reliability of the data was verified by consultation with bank managers, who identified typical rush hours and typical slack times. When combining two or more data sets collected at different times, data reliability can be further enhanced by objective statistical tests for homogeneity of data. (Do two data sets $\{X_i\}$ and $\{Y_i\}$ on service times for personal accounts, collected at two different times, come from the same parent population? If so, the two sets can be combined.) Additional tests may be required to test for correlation in the data. As soon as the analyst is assured of dealing with a random sample (i.e., correlation is not present), the statistical analysis can begin.

The procedures for analyzing input data from a random sample were discussed in detail in Chapter 9. Whether by hand, or using computer software for the purpose, the analysis consists of three steps:

1. Identifying the appropriate probability distribution
2. Estimating the parameters of the hypothesized distribution
3. Validating the assumed statistical model by a goodness-of-fit test, such as the chi-square or Kolmogorov-Smirnov test, and by graphical methods

The use of goodness-of-fit tests is an important part of the validation of the model assumptions.

10.3.3 Validating Input-Output Transformations

The ultimate test of a model, and in fact the only objective test of the model as a whole, is its ability to predict the future behavior of the real system when the model input data match the real inputs and when a policy implemented in the model is implemented at some point in the system. Furthermore, if the level of some input variables (e.g., the arrival rate of customers to a service facility) were to increase or decrease, the model should accurately predict what would happen in the real system under similar circumstances. In other words, the structure of the model should be accurate enough for the model to make good predictions, not just for one input data set, but for the range of input data sets which are of interest.

In this phase of the validation process, the model is viewed as an input–output transformation. That is, the model accepts values of the input parameters and transforms these inputs into output measures of performance. It is this correspondence that is being validated.

Instead of validating the model input-output transformations by predicting the future, the modeler may use past historical data which have been reserved for validation purposes only; that is, if one data set has been used to develop and calibrate the model, it is recommended that a separate data set be used as the final validation test. Thus, accurate "prediction of the past" may replace prediction of the future for the purpose of validating the model.

A model is usually developed with primary interest in a specific set of system responses to be measured under some range of input conditions. For example, in a queueing system, the responses may be server utilization and customer delay, and the range of input conditions (or input variables) may include two or three servers at some station and a choice of scheduling rules. In a production system, the response may be throughput (i.e., production per hour) and the input conditions may be a choice of several machines that run at different speeds, with each machine having its own breakdown and maintenance characteristics. In any case, the modeler should use the main responses of interest as the criteria for validating a model. If the model is used later for a purpose different from its original purpose, the model should be revalidated in terms of the new responses of interest and under the possibly new input conditions.

A necessary condition for the validation of input-output transformations is that some version of the system under study exists, so that system data under at least one set of input conditions can be collected to compare to model predictions. If the system is in the planning stages and no system operating data can be collected, complete input-output validation is not possible. Other types of validation should be conducted, to the extent possible. In some cases, subsystems of the planned system may exist and a partial input-output validation can be conducted.

Presumably, the model will be used to compare alternative system designs, or to investigate system behavior under a range of new input conditions. Assume for now that some version of the system is operating, and that the

model of the existing system has been validated. What, then, can be said about the validity of the model of a nonexistent proposed system, or the model of the existing system under new input conditions?

First, the responses of the two models under similar input conditions will be used as the criteria for comparison of the existing system to the proposed system. Validation increases the modeler's confidence that the model of the existing system is accurate. Second, in many cases, the proposed system is a modification of the existing system, and the modeler hopes that confidence in the model of the existing system can be transferred to the model of the new system. This transfer of confidence usually can be justified if the new model is a relatively minor modification of the old model in terms of changes to the computerized representation of the system (it may be a major change for the actual system). Changes in the computerized representation of the system, ranging from relatively minor to relatively major, include:

1. Minor changes of single numerical parameters, such as the speed of a machine, the arrival rate of customers (with no change in distributional form of interarrival times), or the number of servers in a parallel service center
2. Minor changes of the form of a statistical distribution, such as the distribution of a service time or a time to failure of a machine
3. Major changes in the logical structure of a subsystem, such as a change in queue discipline for a waiting-line model, or a change in the scheduling rule for a job shop model
4. Major changes involving a different design for the new system, such as a computerized inventory control system replacing an older noncomputerized system, or an automatic computerized storage and retrieval system replacing a warehouse system in which workers pick items manually

If the change to the computerized representation of the system is minor, such as in items 1 or 2, these changes can be carefully verified and output from the new model accepted with considerable confidence. If a sufficiently similar subsystem exists elsewhere, it may be possible to validate the submodel that represents the subsystem and then to integrate this submodel with other validated submodels to build a complete model. In this way, partial validation of the substantial model changes in items 3 and 4 may be possible. Unfortunately, there is no way to completely validate the input-output transformations of a model of a nonexisting system. In any case, within time and budget constraints the modeler should use as many validation techniques as possible, including input-output validation of subsystem models if operating data can be collected on such subsystems.

Example 10.2 will illustrate some of the techniques that are possible for input-output validation, and will discuss the concepts of an input variable, uncontrollable variable, decision variable, output or response variable, and input-output transformation in more detail.

EXAMPLE 10.2 (The Fifth National Bank of Jaspar)

The Fifth National Bank of Jaspar, as shown in Figure 10.4, is planning to expand its drive-in service at the corner of Main Street. Currently, there is one drive-in window serviced by one teller. Only one or two transactions are allowed at the drive-in window, so it was assumed that each service time was a random sample from some underlying population. Service times $\{S_i, i = 1, 2, \ldots, 90\}$ and interarrival times $\{A_i, i = 1, 2, \ldots, 90\}$ were collected for the 90 customers who arrived between 11:00 A.M. and 1:00 P.M. on a Friday. This time slot was selected for data collection after consultation with management and the teller because it was felt to be representative of a typical rush hour.

Data analysis (as outlined in Chapter 9) led to the conclusion that arrivals could be modeled as a Poisson process at a rate of 45 customers per hour, and that service times were approximately normally distributed with a mean of 1.1 minutes and a standard deviation of 0.2 minute. Thus, the model has two input variables:

1. Interarrival times, exponentially distributed (i.e., a Poisson arrival process) at rate $\lambda = 45$ per hour
2. Service times, assumed to be $N(1.1, (0.2)^2)$

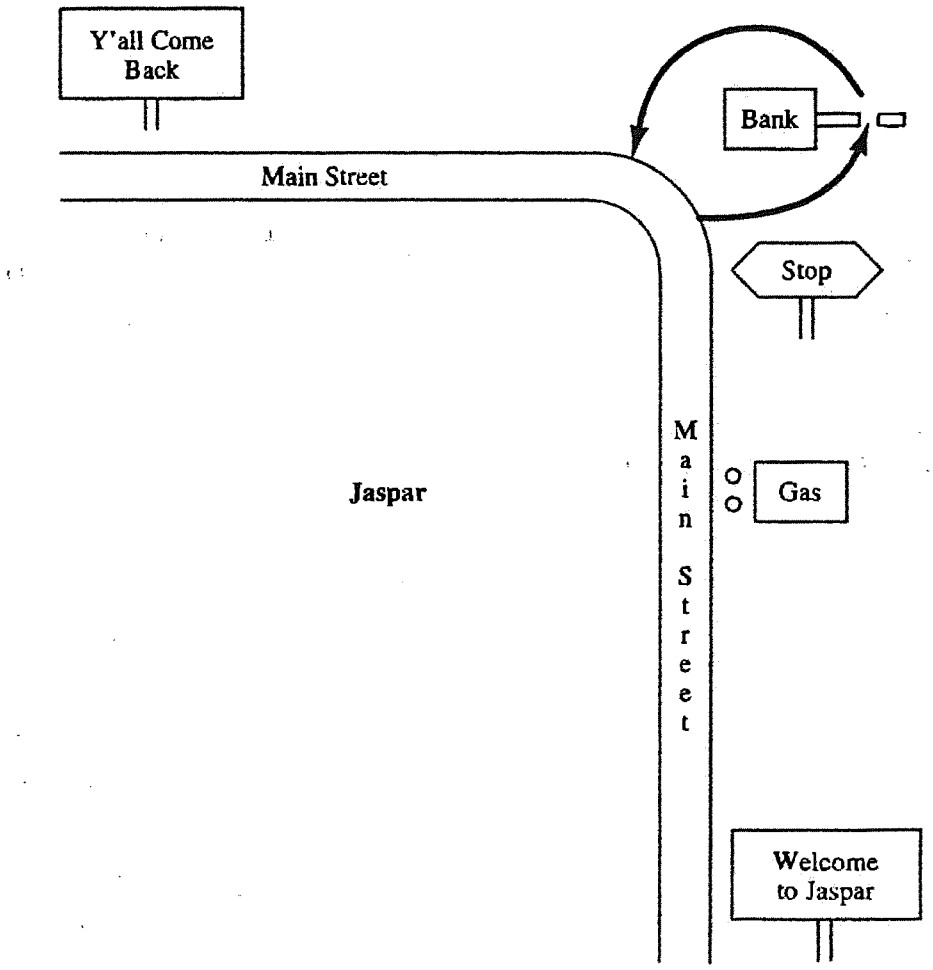

Figure 10.4. Drive-in window at the Fifth National Bank.

Each input variable has a level: the rate ($\lambda = 45$ per hour) for the interarrival times, and the mean 1.1 minutes and standard deviation 0.2 minute for the service times. The interarrival times are examples of uncontrollable variables (i.e., uncontrollable by management in the real system). The service times are also uncontrollable variables, although the level of the service times may be partially controllable. If the mean service time could be decreased to 0.9 minute by increasing the technology, the level of the service-time variable becomes a decision variable or controllable parameter. Setting all decision variables at some level constitutes a policy. For example, the current bank policy is one teller ($D_1 = 1$), mean service time $D_2 = 1.1$ minutes, and one line for waiting cars ($D_3 = 1$). (D_1, D_2, ... are used to denote decision variables.) Decision variables are under management's control; the uncontrollable variables, such as arrival rate and actual arrival times, are not under management's control. The arrival rate may change from time to time, but such change is due to external factors not under management's control.

A model of current bank operations was developed and verified in close consultation with bank management and employees. Model assumptions were validated, as discussed in Section 10.3.2. The resulting model is now viewed as a "black box" which takes all input variable specifications and transforms them into a set of output or response variables. The output variables consist of all statistics of interest generated by the simulation about the model's behavior. For example, management is interested in the teller's utilization at the drive-in window (percent of time the teller is busy at the window), average delay in minutes of a customer from arrival to beginning of service, and the maximum length of the line during the rush hour. These input and output variables are shown in Figure 10.5, and are listed in Table 10.1 together with some additional output variables. The uncontrollable input variables are denoted by X, the decision variables by D, and the output variables by Y. From the "black-box" point of view, the model takes the inputs X and D and produces the outputs Y, namely

$$(X, D) \xrightarrow{f} Y$$

or

$$f(X, D) = Y$$

Here f denotes the transformation that is due to the structure of the model. For the Fifth National Bank study, the exponentially distributed interarrival time generated in the model (by the methods of Chapter 8) between customer $n-1$ and customer n is denoted by X_{1n}. (Do not confuse X_{1n} with A_n; the latter was an observation made on the real system.) The normally distributed service time generated in the model for customer n is denoted by X_{2n}. The set of decision variables, or policy, is $D = (D_1, D_2, D_3) = (1, 1.1, 1)$ for current operations. The output, or response, variables are denoted by $Y = (Y_1, Y_2, \ldots, Y_7)$ and are defined in Table 10.1.

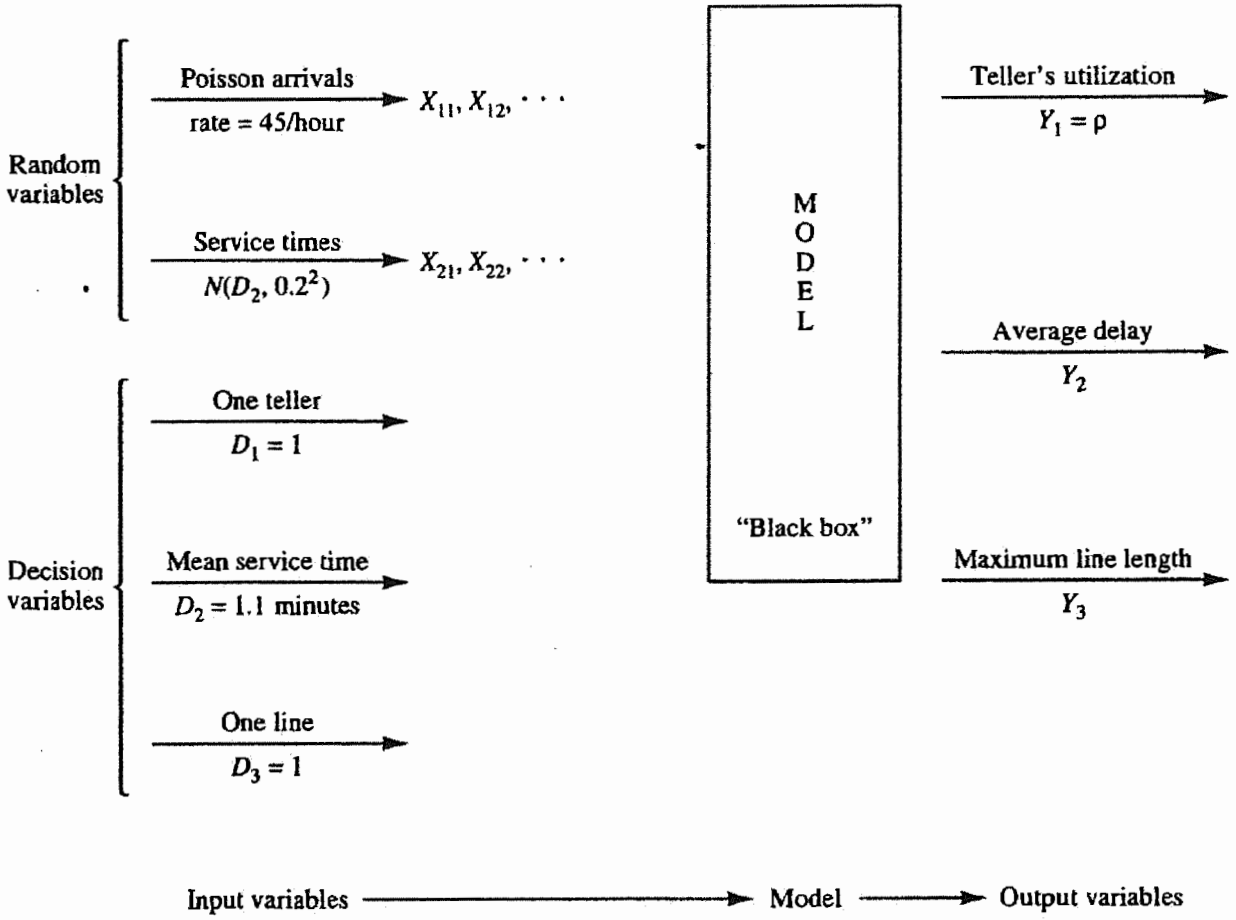

Figure 10.5. Model input-output transformation.

For validation of the input-output transformations of the bank model to be possible, real system data must be available, comparable to at least some of the model output Y of Table 10.1. The system responses should have been collected during the same time period (from 11:00 A.M. to 1:00 P.M. on the

Table 10.1. Input and Output Variables for Model of Current Bank Operations

Input Variables	*Model Output Variables, Y*
D = decision variables	Variables of primary interest
X = other variables	to management (Y_1, Y_2, Y_3):
	Y_1 = teller's utilization
Poisson arrivals at rate = 45/hour	Y_2 = average delay
$X_{11}, X_{12}, \ldots$	Y_3 = maximum line length
[3pt] Service times, $N(D_2, 0.2^2)$	Other output variables of
$X_{21}, X_{22}, \ldots$	secondary interest:
	Y_4 = observed arrival rate
$D_1 = 1$ (one teller)	Y_5 = average service time
$D_2 = 1.1$ minutes (mean service time)	Y_6 = sample standard deviation of service
$D_3 = 1$ (one line)	times
	Y_7 = average length of time

Table 10.2. Results of Six Replications of the First Bank Model

Replication	Y_4 (Arrivals/Hour)	Y_5 (Minutes)	Y_2 = Average Delay (Minutes)
1	51	1.07	2.79
2	40	1.12	1.12
3	45.5	1.06	2.24
4	50.5	1.10	3.45
5	53	1.09	3.13
6	49	1.07	2.38
Sample mean			2.51
Standard deviation			0.82

same Friday) in which the input data $\{A_i, S_i\}$ were collected. This is important because if system response data were collected on a slower day (say, an arrival rate of 40 per hour), the system responses such as teller utilization (Z_1), average delay (Z_2), and maximum line length (Z_3) would be expected to be lower than the same variables during a time slot when the arrival rate was 45 per hour, as observed. Suppose that the delay of successive customers was measured on the same Friday between 11:00 A.M. and 1:00 P.M., and that the average delay was found to be $Z_2 = 4.3$ minutes.

When the model is run using generated random variates X_{1n} and X_{2n}, it is expected that observed values of average delay, Y_2, should be close to $Z_2 = 4.3$ minutes. However, the generated input values (X_{1n} and X_{2n}) cannot be expected to replicate exactly the actual input values (A_n and S_n) of the real system, but they are expected to replicate the statistical pattern of the actual inputs. Hence, simulation-generated values of Y_2 are expected to be consistent with the observed system variable $Z_2 = 4.3$ minutes. Now consider how the modeler might test this consistency.

The modeler makes a small number of statistically independent replications of the model. Statistical independence is guaranteed by using nonoverlapping sets of random numbers produced by the random-number generator, or by choosing seeds for each replication independently (from a random-number table). The results of six independent replications, each of 2 hours duration, are given in Table 10.2.

Observed arrival rate Y_4 and sample average service time Y_5 for each replication of the model are also noted, to be compared to the specified values of 45/hour and 1.1 minutes, respectively. The validation test consists of comparing the system response, namely average delay $Z_2 = 4.3$ minutes, to the model responses, Y_2. Formally, a statistical test of the null hypothesis

$$H_0\text{: } E(Y_2) = 4.3 \text{ minutes}$$

versus (10.1)

$$H_1\text{: } E(Y_2) \neq 4.3 \text{ minutes}$$

is conducted. If H_0 is not rejected, then on the basis of this test there is no reason to consider the model invalid. If H_0 is rejected, the current version of the model is rejected and the modeler is forced to seek ways to improve the model, as illustrated by Figure 10.3. As formulated here, the appropriate statistical test is the t-test, which is conducted in the following manner:

Choose a level of significance α and a sample size n. For the bank model, choose

$$\alpha = 0.05, \quad n = 6$$

Compute the sample mean $\bar{Y}_2$ and the sample standard deviation S over the n replications by Equations (9.1) and (9.2):

$$\bar{Y}_2 = \frac{1}{n}\sum_{i=1}^{n} Y_{2i} = 2.51 \text{ minutes}$$

and

$$S = \left[\frac{\sum_{i=1}^{n}(Y_{2i} - \bar{Y}_2)^2}{n-1}\right]^{1/2} = 0.82 \text{ minute}$$

where Y_{2i}, $i = 1, \ldots, 6$, are as shown in Table 10.2.

Get the critical value of t from Table A.5. For a two-sided test such as that in Equation (10.1), use $t_{\alpha/2,n-1}$; for a one-sided test, use $t_{\alpha,n-1}$ or $-t_{\alpha,n-1}$ as appropriate ($n - 1$ is the degrees of freedom). From Table A.5, $t_{0.025,5} = 2.571$ for a two-sided test.

Compute the test statistic

$$t_0 = \frac{\bar{Y}_2 - \mu_0}{S/\sqrt{n}} \tag{10.2}$$

where μ_0 is the specified value in the null hypothesis, H_0. Here $\mu_0 = 4.3$ minutes, so that

$$t_0 = \frac{2.51 - 4.3}{0.82/\sqrt{6}} = -5.34$$

For the two-sided test, if $|t_0| > t_{\alpha/2,n-1}$, reject H_0. Otherwise, do not reject H_0. [For the one-sided test with H_1: $E(Y_2) > \mu_0$, reject H_0 if $t > t_{\alpha,n-1}$; with H_1: $E(Y_2) < \mu_0$, reject H_0 if $t < -t_{\alpha,n-1}$.]

Since $|t| = 5.34 > t_{0.025,5} = 2.571$, reject H_0 and conclude that the model is inadequate in its prediction of average customer delay.

Recall that when testing hypotheses, rejection of the null hypothesis H_0 is a strong conclusion, because

$$P(H_0 \text{ rejected} \mid H_0 \text{ is true}) = \alpha \tag{10.3}$$

and the level of significance α is chosen small, say $\alpha = 0.05$, as was done here. Equation (10.3) says that the probability of making the error of rejecting H_0 when H_0 is in fact true is low ($\alpha = 0.05$); that is, the probability is small of declaring the model invalid when it is valid (with respect to the variable being

tested). The assumptions justifying a t-test are that the observations (Y_{2i}) are normally and independently distributed. Are these assumptions met in the present case?

1. The ith observation Y_{2i} is the average delay of all drive-in customers who began service during the ith simulation run of 2 hours, and thus by a central-limit-theorem effect, it is reasonable to assume that each observation Y_{2i} is approximately normally distributed, provided that the number of customers it is based on is not too small.
2. The observations Y_{2i}, $i = 1, \ldots, 6$, are statistically independent by design, that is, by choice of the random-number seeds independently for each replication, or by use of nonoverlapping streams.
3. The t-statistic computed by Equation (10.2) is a robust statistic; that is, it is approximately distributed as the t-distribution with $n - 1$ degrees of freedom, even when $Y_{21}, Y_{22}, \ldots$ are not exactly normally distributed, and thus the critical values in Table A.5 can reliably be used.

Now that the model of the Fifth National Bank of Jaspar has been found lacking, what should the modeler do? Upon further investigation, the modeler realized that the model contained two unstated assumptions:

1. When a car arrived to find the window immediately available, the teller began service immediately.
2. There is no delay between one service ending and the next beginning, when a car is waiting.

Assumption 2 was found to be approximately correct because a service time was considered to begin when the teller actually began service but was not considered to have ended until the car had exited the drive-in window and the next car, if any, had begun service, or the teller saw that the line was empty. On the other hand, assumption 1 was found to be incorrect because the teller had other duties—mainly serving walk-in customers if no cars were present—and tellers always finished with a previous customer before beginning service on a car. It was found that walk-in customers were always present during rush hour; that the transactions were mostly commercial in nature, taking a considerably longer time than the time required to service drive-up customers; and that when an arriving car found no other cars at the window, it had to wait until the teller finished with the present walk-in customer. To correct this model inadequacy, the structure of the model was changed to include the additional demand on the teller's time, and data were collected on service times of walk-in customers. Analysis of these data found that they were approximately exponentially distributed with a mean of 3 minutes.

The revised model was run, yielding the results in Table 10.3. A test of the null hypothesis H_0: $E(Y_2) = 4.3$ minutes [as in Equation (10.1)] was again conducted, according to the procedure previously outlined.

Table 10.3. Results of Six Replications of the Revised Bank Model

Replication	Y_4 (Arrivals/Hour)	Y_5 (Minutes)	Y_2 = Average Delay (Minutes)
1	51	1.07	5.37
2	40	1.11	1.98
3	45.5	1.06	5.29
4	50.5	1.09	3.82
5	53	1.08	6.74
6	49	1.08	5.49
Sample mean			4.78
Standard deviation			1.66

Choose $\alpha = 0.05$ and $n = 6$ (sample size).
Compute $\bar{Y}_2 = 4.78$ minutes, $S = 1.66$ minutes.
From Table A.5, the critical value is $t_{0.25,5} = 2.571$.
Compute the test statistic $t_0 = (\bar{Y}_2 - \mu_0)/(S/\sqrt{n}) = 0.710$.
Since $|t_0| < t_{0.025,5} = 2.571$, do not reject H_0, and thus tentatively accept the model as valid.

Failure to reject H_0 must be considered as a weak conclusion unless the power of the test has been estimated and found to be high (close to 1). That is, it can only be concluded that the data at hand $(Y_{21}, \ldots, Y_{26})$ were not sufficient to reject the hypothesis H_0: $\mu_0 = 4.3$ minutes. In other words, this test detects no inconsistency between the sample data $(Y_{21}, \ldots, Y_{26})$ and the specified mean μ_0.

The power of a test is the probability of detecting a departure from H_0: $\mu = \mu_0$ when in fact such a departure exists. In the validation context, the power of the test is the probability of detecting an invalid model. The power may also be expressed as 1 minus the probability of a Type II, or β, error, where $\beta = P(\text{Type II error}) = P(\text{failing to reject } H_0 | H_1 \text{ is true})$ is the probability of accepting the model as valid when it is not valid.

To consider failure to reject H_0 as a strong conclusion, the modeler would want β to be small. Now, β depends on the sample size n and on the true difference between $E(Y_2)$ and $\mu_0 = 4.3$ minutes — that is, on

$$\delta = \frac{|E(Y_2) - \mu_0|}{\sigma}$$

where σ, the population standard deviation of an individual Y_{2i}, is estimated by S. Tables A.10 and A.11 are typical operating-characteristic (OC) curves, which are graphs of the probability of a Type II error $\beta(\delta)$ versus δ for given sample size n. Table A.10 is for a two-sided t-test while Table A.11 is for a one-sided t-test. Suppose that the modeler would like to reject H_0 (model validity) with probability at least 0.90 if the true mean delay of the model, $E(Y_2)$, differed from the average delay in the system, $\mu_0 = 4.3$ minutes, by 1

minute. Then δ is estimated by

$$\hat{\delta} = \frac{|E(Y_2) - \mu_0|}{S} = \frac{1}{1.66} = 0.60$$

For the two-sided test with $\alpha = 0.05$, use of Table A.10 results in

$$\beta(\hat{\delta}) = \beta(0.6) = 0.75 \text{ for } n = 6$$

To guarantee that $\beta(\hat{\delta}) \leq 0.10$, as was desired by the modeler, Table A.10 reveals that a sample size of approximately $n = 30$ independent replications would be required. That is, for a sample size $n = 6$ and assuming that the population standard deviation is 1.66, the probability of accepting H_0 (model validity), when in fact the model is invalid ($|E(Y_2) - \mu_0| = 1$ minute), is $\beta = 0.75$, which is quite high. If a 1-minute difference is critical, and if the modeler wants to control the risk of declaring the model valid when model predictions are as much as 1 minute off, a sample size of $n = 30$ replications is required to achieve a power of 0.9. If this sample size is too high, either a higher β risk (lower power) or a larger difference δ must be considered. ◀

In general, it is always best to control the Type II error, or β error, by specifying a critical difference δ and choosing a sample size by making use of an appropriate OC curve. (Computation of power and use of OC curves for a wide range of tests is discussed in Hines and Montgomery [1990].) In summary, in the context of model validation, the Type I error is the rejection of a valid model and is easily controlled by specifying a small level of significance α (say $\alpha = 0.2, 0.1, 0.05$, or 0.01). The Type II error is the acceptance of a model as valid when it is invalid. For a fixed sample size n, increasing α will decrease β, the probability of a Type II error. Once α is set, and the critical difference to be detected is selected, the only way to decrease β is to increase the sample size. A Type II error is the more serious of the two types of errors, and thus it is important to design the simulation experiments to control the risk of accepting an invalid model. The two types of error are summarized in Table 10.4, which compares statistical terminology to modeling terminology.

Note that validation is not to be viewed as an either/or proposition, but rather should be viewed in the context of calibrating a model, as conceptually exhibited in Figure 10.3. If the current version of the bank model produces estimates of average delay (Y_2) that are not close enough to real system behavior

Table 10.4. Types of Error in Model Validation

Statistical Terminology	*Modeling Terminology*	*Associated Risk*
Type I: rejecting H_0 when H_0 is true	Rejecting a valid model	α
Type II: failure to reject H_0 when H_1 is true	Failure to reject an invalid model	β

($\mu_0 = 4.3$ minutes), the source of the discrepancy is sought, and the model is revised in light of this new knowledge. This iterative process is repeated until model accuracy is judged adequate.

10.3.4 Input-Output Validation: Using Historical Input Data

When using artificially generated data as input data, as was done to test the validity of the bank models in Section 10.3.3, the modeler expects the model to produce event patterns that are compatible with, but not identical to, the event patterns that occurred in the real system during the period of data collection. Thus, in the bank model, artificial input data $\{X_{1n}, X_{2n}, n = 1, 2, \ldots\}$ for interarrival and service times were generated and replicates of the output data Y_2 were compared to what was observed in the real system by means of the hypothesis test stated in Equation (10.1). An alternative to generating input data is to use the actual historical record, $\{A_n, S_n, n = 1, 2, \ldots\}$, to drive the simulation model and then to compare model output to system data.

To implement this technique for the bank model, the data $A_1, A_2, \ldots$ and $S_1, S_2, \ldots$ would have to be entered into the model into arrays, or stored on a file to be read as the need arose. Just after customer n arrived at time $t_n = \sum_{i=1}^{n} A_i$, customer $n + 1$ would be scheduled on the future event list to arrive at future time $t_n + A_{n+1}$ (without any random numbers being generated). If customer n were to begin service at time t_n', a service completion would be scheduled to occur at time $t_n' + S_n$. This event scheduling without random-number generation could be implemented quite easily in most simulation languages by using arrays to store the data.

When using this technique, the modeler hopes that the simulation will duplicate as closely as possible the important events that occurred in the real system. In the model of the Fifth National Bank of Jaspar, the arrival times and service durations will exactly duplicate what happened in the real system on that Friday between 11:00 A.M. and 1:00 P.M. If the model is sufficiently accurate, then the delays of customers, lengths of lines, utilizations of servers, and departure times of customers predicted by the model will be close to what actually happened in the real system. It is, of course, the model builder's and model user's judgment that determines the level of accuracy required.

To conduct a validation test using historical input data, it is important that all the input data ($A_n, S_n, \ldots$) and all the system response data, such as average delay (Z_2), be collected during the same time period. Otherwise, the comparison of model responses to system responses, such as the comparison of average delay in the model (Y_2) to that in the system (Z_2), could be misleading. The responses (Y_2 and Z_2) depend on the inputs (A_n and S_n) as well as on the structure of the system, or model. Implementation of this technique could be difficult for a large system because of the need for simultaneous data collection of all input variables and those response variables of primary interest. In some systems, electronic counters and devices are used to ease the data-collection task by automatically recording certain types of data. The following example

was based on two simulation models reported in Carson et al. [1981a, b], in which simultaneous data collection and the subsequent validation were both completed suceessfully.

EXAMPLE 10.3 (The Candy Factory)

The production line at the Sweet Li'l Things Candy Factory in Decatur consists of three machines which make, package, and box their famous candy. One machine (the candy maker) makes and wraps individual pieces of candy and sends them by conveyor to the packer. The second machine (the packer) packs the individual pieces into a box. A third machine (the box maker) forms the boxes and supplies them by conveyor to the packer. The system is illustrated in Figure 10.6.

Each machine is subject to random breakdowns due to jams and other causes. These breakdowns cause the conveyor to begin to empty or fill. The conveyors between the two makers and the packer are used as a temporary storage buffer for in-process inventory. In addition to the randomly occurring breakdowns, if the candy conveyor empties, a packer runtime is interrupted and the packer remains idle until more candy is produced. If the box conveyor empties because of a long random breakdown of the box machine, an operator manually places racks of boxes onto the packing machine. If the conveyor fills, the corresponding maker becomes idle. The purpose of the model is to investigate the frequency of these operator interventions which require manual loading of racks of boxes, as a function of various combinations of individual machines and lengths of conveyor. Different machines have different production speeds and breakdown characteristics, and longer conveyors can hold more in-process inventory. The goal is to hold operator interventions to an acceptable level while maximizing production. As machine stoppages (due to a full or empty conveyor) cause increased damage to the product, this is also a factor in production.

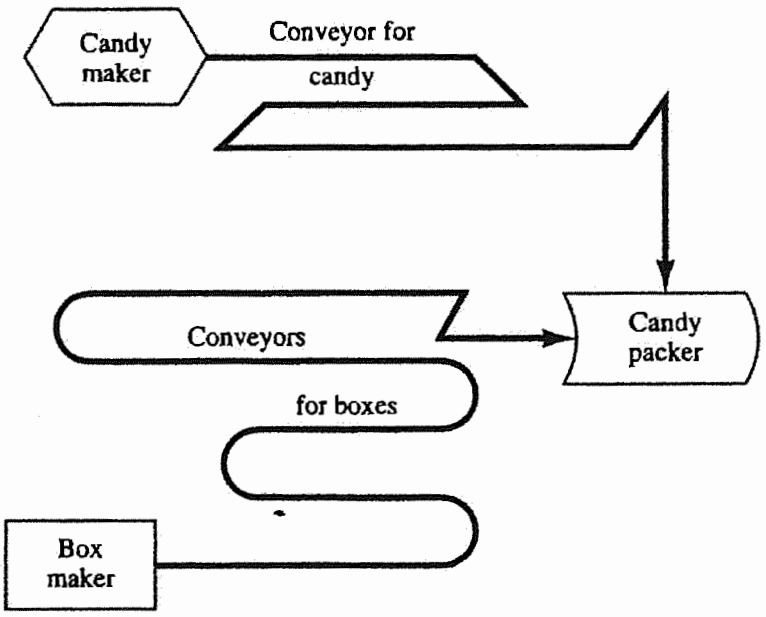

Figure 10.6. Production line at the candy factory.

Table 10.5. Validation of the Candy Factory Model

Response, i	*System, Z_i*	*Model, Y_i*
1. Production level	897, 208	883, 150
2. Number of operator interventions	3	3
3. Time of occurrence	7:22, 8:41, 10:10	7:24, 8:42, 10:14

A simulation model of the Candy Factory was developed and a validation effort using historical inputs was conducted. Engineers in the Candy Factory set aside a 4-hour time slot from 7:00 A.M. to 11:00 A.M. to collect data on an existing production line. For each machine, say machine i, time to failure and random downtime data

$$T_{i1}, D_{i1}, T_{i2}, D_{i2}, \ldots$$

were collected. For machine $i (i = 1, 2, 3)$, T_{ij} is the jth runtime (or time to failure), and D_{ij} is the successive random downtime. A runtime, T_{ij}, can be interrupted due to a full or empty conveyor (as appropriate) but resumes when conditions are right. Initial system conditions at 7:00 A.M. were recorded so that they could be duplicated in the model as initial conditions at time 0. Additionally, system responses of primary interest—the production level (Z_1) and the number (Z_2) and time of occurrence (Z_3) of operator interventions—were recorded for comparison with model predictions.

The system input data, T_{ij} and D_{ij}, were fed into the model and used as runtimes and random downtimes. The structure of the model determined the occurrence of shutdowns due to a full or empty conveyor, and the occurrence of operator interventions. Model response variables ($Y_i, i = 1, 2, 3$) were collected for comparison to the corresponding system response variables ($Z_i, i = 1, 2, 3$).

The closeness of model predictions to system performance aided the engineering staff considerably in convincing management of the validity of the model. These results are shown in Table 10.5. A simple display such as Table 10.5 can be quite effective in convincing skeptical engineers and managers of a model's validity—perhaps more effective than the most sophisticated statistical methods! ◀

With only one set of historical input and output data, only one set of simulated output data can be obtained, and thus no simple statistical tests are possible based on summary measures. But if K historical input data sets are collected, and K observations $Z_{i1}, Z_{i2}, \ldots, Z_{iK}$ of some system response variable, Z_i, are collected, such that the output measure Z_{ij} corresponds to the jth input set, an objective statistical test becomes possible. For example, Z_{ij} could be the average delay of all customers who were served during the time the jth input data set was collected. With the K input data sets in hand, the modeler now runs the model K times, once for each input set, and observes the simulated results $W_{i1}, W_{i2}, \ldots, W_{iK}$ corresponding to $Z_{ij}, j = 1, \ldots, K$.

Table 10.6. Comparison of System and Model Output Measures When Using Identical Historical Inputs

Input Data Set	*System Output,* Z_{ij}	*Model Output,* W_{ij}	*Observed Difference,* d_j	*Squared Deviation from Mean,* $(d_j - \bar{d})^2$
1	Z_{i1}	W_{i1}	$d_1 = Z_{i1} - W_{i1}$	$(d_1 - \bar{d})^2$
2	Z_{i2}	W_{i2}	$d_2 = Z_{i2} - W_{i2}$	$(d_2 - \bar{d})^2$
3	Z_{i3}	W_{i3}	$d_3 = Z_{i3} - W_{i3}$	$(d_3 - \bar{d})^2$
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
K	Z_{iK}	W_{iK}	$d_K = Z_{iK} - W_{iK}$	$(d_K - \bar{d})^2$
			$\bar{d} = \frac{1}{K}\sum_{j=1}^{K} d_j$	$S_d^2 = \frac{1}{K-1}\sum_{j=1}^{K}(d_j - \bar{d})^2$

Continuing the same example, W_{ij} would be the average delay predicted by the model when using the jth input set. The available data for comparison appears as in Table 10.6.

If the K input data sets are fairly homogeneous, it is reasonable to assume that the K observed differences $d_j = Z_{ij} - W_{ij}$, $j = 1, \ldots, K$, are identically distributed. Furthermore, if the collection of the K sets of input data was separated in time, say on different days, it is reasonable to assume that the K differences $d_1, \ldots, d_K$ are statistically independent, and hence the differences $d_1, \ldots, d_K$ constitute a random sample. In many cases, each Z_i and W_i is a sample average over customers, so that (by the central limit theorem) the differences $d_j = Z_{ij} - W_{ij}$ are approximately normally distributed with some mean μ_d and variance σ_d^2. The appropriate statistical test is then a t-test of the null hypothesis of no mean difference:

$$H_0\colon \mu_d = 0$$

versus the alternative of significant difference:

$$H_1\colon \mu_d \neq 0$$

The proper test is a paired t-test (Z_{i1} is paired with W_{i1}, since each was produced by the first input data set, and so on). First, compute the sample mean difference $\bar{d}$, and the sample variance S_d^2 by the formulas given in Table 10.6. Then compute the t-statistic by

$$t_0 = \frac{\bar{d} - \mu_d}{S_d/\sqrt{K}} \tag{10.4}$$

(with $\mu_d = 0$), and get the critical value $t_{\alpha/2,K-1}$ from Table A.5, where α is the prespecified significance level and $K - 1$ is the number of degrees of freedom. If $|t_0| > t_{\alpha/2,K-1}$, reject the hypothesis H_0 of no mean difference and conclude

that the model is inadequate. If $|t_0| \leq t_{\alpha/2,K-1}$, do not reject H_0 and hence conclude that this test provides no evidence of model inadequacy.

EXAMPLE 10.4 (The Candy Factory, Continued)

Engineers at the Sweet Li'l Things Candy Factory decided to expand the initial validation effort reported in Example 10.3. Electronic devices were installed which could automatically monitor one of the production lines, and the validation effort of Example 10.3 was repeated with $K = 5$ sets of input data. The system and the model were compared on the basis of production level. The results are shown in Table 10.7.

Table 10.7. Validation of the Candy Factory Model (Continued)

Input Data Set, j	*System Production, Z_{1j}*	*Model Production, W_{1j}*	*Observed Difference, d_j*	*Squared Deviation from Mean, $(d_j - \bar{d})^2$*
1	897,208	883,150	14,058	7.594×10^7
2	629,126	630,550	−1,424	4.580×10^7
3	735,229	741,420	−6,191	1.330×10^8
4	797,263	788,230	9,033	1.362×10^7
5	825,430	814,190	11,240	3.4772×10^7
			$\bar{d} = 5{,}343.2$	$S_d^2 = 7.580 \times 10^7$

A paired t-test was conducted to test H_0: $\mu_d = 0$, or equivalently, H_0: $E(Z_1) = E(W_1)$, where Z_1 is the system production level and W_1 is the production level predicted by the simulated model. Let the level of significance be $\alpha = 0.05$. Using the results in Table 10.7, the test statistic, as given by Equation (10.4), is

$$t_0 = \frac{\bar{d}}{S_d/\sqrt{K}} = \frac{5343.2}{8705.85/\sqrt{5}} = 1.37$$

From Table A.5, the critical value is $t_{\alpha/2,K-1} = t_{0.025,4} = 2.78$. Since $|t_0| = 1.37 < t_{0.025,4} = 2.78$, the null hypothesis cannot be rejected on the basis of this test; that is, no inconsistency is detected between system response and model predictions in terms of mean production level. If H_0 had been rejected, the modeler would have searched for the cause of the discrepancy and revised the model, in the spirit of Figure 10.3. ◀

10.3.5 Input-Output Validation: Using a Turing Test

In addition to statistical tests, or when no statistical test is readily applicable, persons knowledgeable about system behavior can be used to compare model output to system output. For example, suppose that five reports of system performance over five different days are prepared, and simulation output data are used to produce five "fake" reports. The 10 reports should all be in exactly

the same format and should contain information of the type that managers and engineers have previously seen on the system. The 10 reports are randomly shuffled and given to the engineer, who is asked to decide which reports are fake and which are real. If the engineer identifies a substantial number of the fake reports, the model builder questions the engineer and uses the information gained to improve the model. If the engineer cannot distinguish between fake and real reports with any consistency, the modeler will conclude that this test provides no evidence of model inadequacy. For further discussion and an application to a real simulation, the reader is referred to Schruben [1980]. This type of validation test is commonly called a Turing test. Its use as model development proceeds can be a valuable tool in detecting model inadequacies, and eventually in increasing model credibility as the model is improved and refined.

10.4 Summary

Validation of simulation models is of great importance. Decisions are made on the basis of simulation results; thus, the accuracy of these results should be subject to question and investigation.

Quite often simulations appear realistic on the surface because simulation models, unlike analytic models, can incorporate any level of detail about the real system. To avoid being "fooled" by this apparent realism, it is best to compare system data to model data, and to make the comparison using a wide variety of techniques, including an objective statistical test, if at all possible.

As discussed by Van Horn [1969, 1971], some of the possible validation techniques, in order of increasing cost-to-value ratios, include:

1. Develop models with high face validity by consulting persons knowledgeable about system behavior on model structure, model input, and model output. Use any existing knowledge in the form of previous research and studies, observation, and experience.
2. Conduct simple statistical tests of input data for homogeneity, randomness, and goodness-of-fit to assumed distributional forms.
3. Conduct a Turing test. Have knowledgeable people (engineers, managers) compare model output to system output and attempt to detect the difference.
4. Compare model output to system output by means of statistical tests.
5. After model development, collect new system data and repeat techniques 2 to 4.
6. Build the new system (or redesign the old one) based on simulation results, collect data on the new system, and use this data to validate the model. (Not recommended if this is the only technique used.)
7. Do little or no validation. Implement simulation results without validating. (Not recommended.)

It is usually too difficult, too expensive, or too time consuming to use all possible validation techniques for every model that is developed. It is an important part of the model builder's task to choose those validation techniques that are most appropriate, both to assure model accuracy and to assure model credibility.

REFERENCES

BALCI, O. [1994], "Validation, Verification and Testing Techniques throughout the Life Cycle of a Simulation Study," *Annals of Operations Research*, Vol. 53, pp. 121–174.

BALCI, O. [1998] "Verification, Validation, and Testing," in *Handbook of Simulation*, ed. J. Banks, John Wiley, New York.

BALCI, O., AND R. G. SARGENT [1982a], "Some Examples of Simulation Model Validation Using Hypothesis Testing," *Proceedings of the Winter Simulation Conference*, ed. H. J. Highland, Y. W. Chao, and O. S. Madrigal, pp. 620–629, Association for Computing Machinery, New York.

BALCI, O., AND R. G. SARGENT [1982b], "Validation of Multivariate Response Models Using Hotelling's Two-Sample T^2 Test," *Simulation*, Vol. 39, No. 6 (Dec.), pp. 185–192.

BALCI, O., AND R. G. SARGENT [1984a], "Validation of Simulation Models via Simultaneous Confidence Intervals," *American Journal of Mathematical Management Sciences*, Vol. 4, Nos. 3 and 4, pp. 375–406.

BALCI, O., AND R. G. SARGENT [1984b], "A Bibliography on the Credibility Assessment and Validation of Simulation and Mathematical Models," *Simuletter*, Vol. 15, No. 3, pp. 15–27.

BORTSCHELLER, B. J., AND E. T. SAULNIER [1992], "Model Reusability in a Graphical Simulation Package," *Proceedings of the Winter Simulation Conference*, ed. J. J. Swain, D. Goldsman, R. C. Crain, J. R. Wilson, pp. 764–772, Association for Computing Machinery, New York.

CARSON, J. S., N. WILSON, D. CARROLL, AND C. H. WYSOWSKI [1981a], "A Discrete Simulation Model of a Cigarette Fabrication Process," *Proceedings of the Twelfth Modeling and Simulation Conference*, University of Pittsburgh.

CARSON, J. S., N. WILSON, D. CARROLL, AND C. H. WYSOWSKI [1981b], "Simulation of a Filter Rod Manufacturing Process," *Proceedings of the 1981 Winter Simulation Conference*, ed. T. I. Oren, C. M. Delfosse, and C. M. Shub, pp. 535–541, Association for Computing Machinery, New York.

CARSON, J. S., [1986], "Convincing Users of Model's Validity Is Challenging Aspect of Modeler's Job," *Industrial Engineering*, June, pp. 76–85.

GAFARIAN, A. V., AND J. E. WALSH [1970], "Methods for Statistical Validation of a Simulation Model for Freeway Traffic near an On-Ramp," *Transportation Research*, Vol. 4, p. 379–384.

GASS, S. I. [1983], "Decision-Aiding Models: Validation, Assessment, and Related Issues for Policy Analysis," *Operations Research*, Vol. 31, No. 4, pp. 601–663.

HINES, W. W., AND D. C. MONTGOMERY [1990], *Probability and Statistics in Engineering and Management Science*, 3d ed., John Wiley, New York.

KLEIJNEN, J. P. C. [1987], *Statistical Tools for Simulation Practitioners*, Marcel Dekker, New York.

KLEIJNEN, J. P. C. [1993] "Simulation and Optimization in Production Planning: A Case Study," *Decision Support Systems*, Vol. 9, pp. 269–280.

KLEIJNEN, J. P. C. [1995], "Theory and Methodology: Verification and Validation of Simulation Models," *European Journal of Operational Research*, Vol. 82, No. 1, pp. 145–162.

LAW, A. M. AND W. D. KELTON [2000], *Simulation Modeling and Analysis*, 3d ed., McGraw-Hill, New York.

NAYLOR, T. H., AND J. M. FINGER [1967], "Verification of Computer Simulation Models," *Management Science*, Vol. 2, pp. B92–B101.

OREN, T. [1981], "Concepts and Criteria to Assess Acceptability of Simulation Studies: A Frame of Reference," *Communications of the Association for Computing Machinery*, Vol. 24, No. 4, pp. 180–89.

SARGENT, R. G. [1994], "Verification and Validation of Simulation Models," *Proceedings of the Winter Simulation Conference*, ed. A. Seila, S. Manivannan, J. Tew, D. Sadowski, pp. 77–87, Association for Computing Machinery, New York.

SCHECTER, M., AND R. C. LUCAS [1980], "Validating a Large Scale Simulation Model of Wilderness Recreation Travel," *Interfaces*, Vol. 10, pp. 11–18.

SCHRUBEN, L. W. [1980], "Establishing the Credibility of Simulations," *Simulation*, Vol. 34, pp. 101–105.

SHANNON, R. E. [1975], *Systems Simulation: The Art and Science*. Prentice Hall, Upper Saddle River, NJ.

VAN HORN, R. L. [1969], "Validation," in *The Design of Computer Simulation Experiments*, ed. T. H. Naylor, Duke University Press, Durham, NC.

VAN HORN, R. L. [1971], "Validation of Simulation Results," *Management Science*, Vol. 17, pp. 247–258.

YOUNGBLOOD, S. M. [1993] "Literature Review and Commentary on the Verification, Validation and Accreditation of Models," Johns Hopkins University, Laurel, MD.

YÜCESAN, E., AND S. H. JACOBSON [1992], "Building Correct Simulation Models is Difficult," *Proceedings of the Winter Simulation Conference*, ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson, pp. 783–790, Association for Computing Machinery, New York.

EXERCISES

1. A simulation model of a job shop was developed to investigate different scheduling rules. To validate the model, the currently used scheduling rule was incorporated into the model and the resulting output compared to observed system behavior. By searching the previous year's computerized records it was estimated that the average number of jobs in the shop was 22.5 on a given day. Seven independent replications of the model were run, each of 30 days duration, with the following results for average number of jobs in the shop:

18.9 22.0 19.4 22.1 19.8 21.9 20.2

(a) Develop and conduct a statistical test to determine if model output is consistent with system behavior. Use a level of significance of $\alpha = 0.05$.

(b) What is the power of this test if a difference of two jobs is viewed as critical? What sample size is needed to guarantee a power of 0.8 or higher? (Use $\alpha = 0.05$.)

2. System data for the job shop of Exercise 1 revealed that the average time spent by a job in the shop was approximately 4 working days. The model made the following predictions on seven independent replications, for average time spent in the shop:

3.70 4.21 4.35 4.13 3.83 4.32 4.05

(a) Is model output consistent with system behavior? Conduct a statistical test using a level of significance $\alpha = 0.01$.

(b) If it is important to detect a difference of 0.5 day, what sample size is needed to have a power of 0.90? Interpret your results in terms of model validity or invalidity. (Use $\alpha = 0.01$.)

3. For the job shop of Exercise 1, four sets of input data were collected over four different 10-day periods, together with the average number of jobs in the shop (Z_i) for each period. The input data were used to drive the simulation model for four runs of 10 days each, and model predictions of average number of jobs in the shop (Y_i) were collected, with these results:

i	1	2	3	4
Z_i	21.7	19.2	22.8	19.4
Y_i	24.6	21.1	19.7	24.9

(a) Conduct a statistical test to check the consistency of system output and model output. Use a level of significance of $\alpha = 0.05$.

(b) If a difference of two jobs is viewed as important to detect, what sample size is required to guarantee a probability of at least 0.80 of detecting this difference if it indeed exists? (Use $\alpha = 0.05$.)

4. Obtain at least two of the papers or reports listed in the References dealing with validation and verification. Write a short essay comparing and contrasting the various philosophies and approaches to the topic of verification and validation.

5. Find several examples of actual simulations reported in the literature in which the authors discuss validation of their model. Is enough detail given to judge the adequacy of the validation effort? If so, compare the reported validation to the criteria set forth in this chapter. Did the authors use any validation technique not discussed in this chapter? [Several potential sources of articles on simulation applications include the two journals *Interfaces* and *Simulation*, and the *Winter Simulation Conference Proceedings*.]

6. Compare and contrast the various simulation languages in their capability to aid the modeler in the often arduous task of debugging and verification (articles discussing the nature of simulation languages may be found in the *Winter Simulation Conference Proceedings*).

7. **(a)** Compare validation in simulation to the validation of theories in the physical sciences.

(b) Compare the issues involved and the techniques available for validation of models of physical systems versus models of social systems.

(c) Contrast the difficulties, and compare the techniques, in validating a model of a manually operated warehouse versus a model of an automated storage and retrieval system.

(d) Repeat (c) for a model of a production system involving considerable manual labor and human decision making, versus a model of the same production system after it has been automated.