

- One of the most important and difficult tasks facing a model developer is the verification and validation of the simulation model.
- It is the job of the model developer to work closely with the end users throughout the period (development and validation to reduce this skepticism and to increase the credibility.)
- **The goal of the validation process is:**
 - To **produce a model** that represents true behavior of the system closely enough for decision-making purposes.
 - **To increase the model's credibility** to an acceptable level to be used by managers and other decision makers
- **Validation is an integral part of model development**
 - **Verification** – building the model correctly (correctly implemented with good input and structure)
 - **Validation** – building the correct model (an accurate representation of the real system)
 - Usually achieved through calibration.

8.1 Model building, verification, and validation

1) First step in model building is observing the real system

- a) Interactions of components, collecting data
- b) Take advantage of people with special knowledge

2) 2nd step: Construct a conceptual model

- a) Assumptions about components - hypotheses
- b) Structure of the system

3) 3rd step: Implementation of an operational model using software

- a) The third step is the translation of the operational model into a computer recognizable form-the computerized model.

Not a linear process.

Will return to each step many times while building, verifying and validating the model

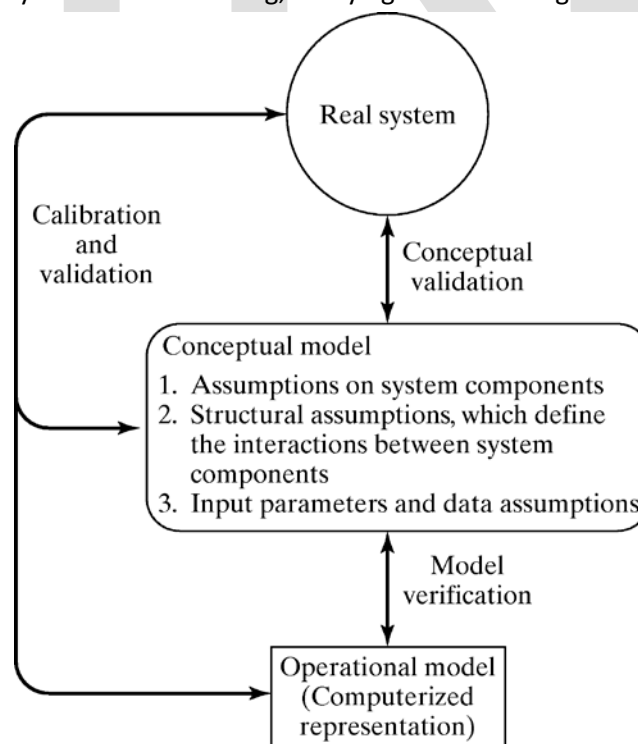


Figure 1 Model building, verification, and validation

8.2 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.

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** that includes each logically possible action a system can take when an event occurs.
3. **Closely examine the model output** for reasonableness under a variety of settings of input parameters.
4. **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 operational model as **self-documenting** as possible.
6. If the operational model is animated, **verify that what is seen in the animation imitates** the actual system.
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 follow ways:

- A. The simulation can be monitored as it progresses.
- B. Attention can be focused on a particular line of logic or multiple lines of logic that constitute a procedure or a particular entity.
- 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 & validation.

8.3 Calibration and Validation of Models

- Verification and validation although are 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 to the model, comparing again and so on.
- The following figure 2 shows the **relationship of the model calibration to the overall validation process**.
- The comparison of the model to reality is carried out by variety of test.
- **Some tests are subjective and others are objective.**
 - **Subjective test** usually **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**.
- **No model is ever a perfect representation of the system**
 - The modeler must weigh the possible, but not guaranteed, increase in model accuracy versus the cost of increased validation effort.
- **Danger during the calibration phase**

Typically few data sets are available, in the worst case only one, and the model is only validated for these.

Solution: If possible collect new data sets.

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

- **Three-step approach:**

- Build a model that has **high face validity**.
- **Validate model assumptions**.
- Compare the model input-output transformations with the real system's data.

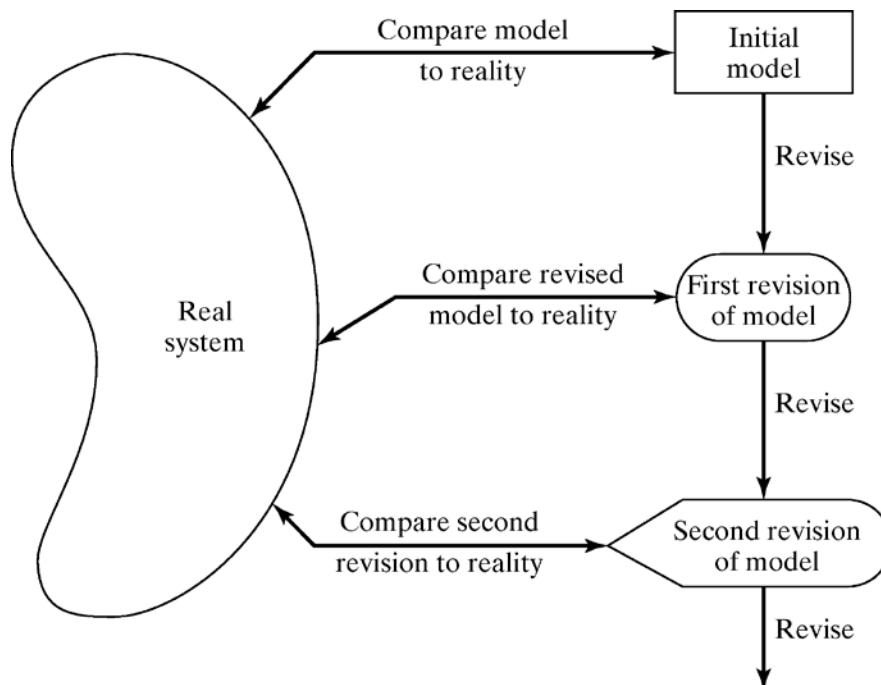


Figure 2 Iterative process of calibration a model

High 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 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.
- Another advantage of user involvement is the increase in the models perceived validity or credibility without which manager will not be willing to trust simulation results as the basis for decision making.
- Sensitivity analysis can also be used to check 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.
- Based on experience and observations on the real system 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.
- 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.

Validate Model Assumptions

- **General classes of model assumptions:**
 - **Structural assumptions:** how the system operates.
 - **Data assumptions:** reliability of data and its statistical analysis.
- Bank example: customer queueing and service facility in a bank.
 - **Structural assumptions**, e.g., customer waiting in one line versus many lines, served FCFS versus priority.

- **Data assumptions**, e.g., interarrival time of customers, service times for commercial accounts.
 - Verify data reliability with bank managers.
 - Test correlation and goodness of fit for data
 - Identify an appropriate probability distribution
 - Estimate the parameters of the hypothesized distribution
 - Validate the assumed statistical model by goodness-of-fit test, such as the chi-square or Kolmogorov-Smirnov test, and by graphical methods

Validating Input-Output Transformation

- In this phase of validation process the model is **viewed as input –output transformation**. That is, the model accepts the values of input parameters and transforms these inputs into output measure of performance. It is this correspondence that is being validated.
- **Instead of validating the model input-output transformation by predicting the future**, the modeler may **use past historical data** which has been served for validation purposes that is, if one set has been used to develop calibrate the model, its recommended that a separate data test be used as final validation test. Thus accurate “prediction of the past” may replace prediction of the future for purpose of validating the future.
- A necessary condition for input-output transformation is that some **version of the system under study exists** so that the system data under at least one set of input condition can be collected to compare to model prediction.
- If the system is in planning stage and **no system operating data** can be collected, complete input-output validation is not possible.
- **Validation increases modeler’s confidence that the model of existing system is accurate.**

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 speed of the machine, arrival rate of the customer etc.
2. Minor changes of the form of a **statistical distribution** such as distribution of service time or a time to failure of a machine.
3. Major changes in the **logical structure of a subsystem** such as change in queue discipline for 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 computerized inventory control system replacing a non computerized system.

Input-Output Validation: Using Historical Input Data

When using artificially generated data as input data the modeler expects the model 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 $\{X_{1n}, X_{2n}, n = 1, 2, \dots\}$ for inter arrival and service times were generated and replicates of the output data Y_2 were compared to what was observed in the real system.
- An alternative to generating input data is to use the actual historical record, $\{A_n, S_n, n = 1, 2, \dots\}$, to drive simulation model and then to compare model output to system data.
- To implement this technique for the bank model, the data $A_1, A_2, \dots, S_1, S_2$ would have to be entered into the model into arrays, or stored on a file to be read as the need arose.
- To conduct a validation test using historical input data, it is important that all input data (A_n, S_n, \dots) and all the system response data, such as average delay (Z_2) , be collected during the same time period.
- Otherwise, 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.

- 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.

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 outputs are used to produce five "fake" reports. The 10 reports should all be in exactly in the same format and should contain information of the type that manager and engineer have previously seen on the system.
 - The ten reports are randomly shuffled and given to the engineers, who are asked to decide which reports are fake and which are real.
 - If engineer identifies substantial number of 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.
 - This type of validation test is called as **TURING TEST.**

8.4 Optimization via simulation

- Optimization via simulation to refer to the problem of maximizing or minimizing the expected (long-run average) performance of a discrete event, stochastic system that is represented by a computer simulation model.
- Optimization usually deals with problems with certainty, but in stochastic discrete-event simulation, the result of any simulation run is a random variable.

Let x_1, x_2, \dots, x_m be the m controllable design variables and $Y(x_1, x_2, \dots, x_m)$ be the observed simulation output performance on one run:

To optimize $Y(x_1, x_2, \dots, x_m)$ with respect to x_1, x_2, \dots, x_m is to maximize or minimize the mathematical expectation (long-run average) of performance.

$$E[Y(x_1, x_2, \dots, x_m)]$$

FAQ's

June 2012

- | | |
|---|------|
| 1) Explain with a neat diagram verification of simulation model. | 10 M |
| 2) Describe with a neat diagram iterative process of calibrating a model. Which are three steps that aid in the validation process? | 10 M |

June 2010

- | | |
|--|------|
| 3) Explain with a neat diagram model building, verification and validation process | 10 M |
| 4) Describe the three steps approach to validation by Naylor and Finger. | 10 M |

Dec 2011

- | | |
|---|------|
| 5) Explain with a neat diagram model building, verification and validation. | 10 M |
|---|------|

June 2011

- | | |
|--|-----|
| 6) Write short note on Optimization via simulation | 5 M |
|--|-----|