

## Systems Simulation Chapter 10: Verification and Validation

Fatih Cavdur  
fatihcavdur@uludag.edu.tr

May 12, 2014

### Introduction

- One of the most important and difficult tasks facing a model developer is the verification and validation of the model.
- System Analyst should work closely with the end users throughout the period of development and validation to increase the model's credibility.
- The goal of the validation process is twofold:
  - To produce a model that represents true system behavior closely enough for the model to be used as a substitute for the actual system.
  - To increase to an acceptable level the credibility of the model, so that the model will be used by managers and other decision makers.

## Introduction

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:

- Verification is concerned with building the model correctly. It proceeds by the comparison of the conceptual model to the computer representation that implements that conception.
- Validation is concerned with building the correct model. It attempts to confirm that a model is an accurate representation of the real system.

## Model Building Process

- The first step in model building consists of observing the real system and the interactions among their various components and of collecting data on their behavior.
- The second step is the construction of a conceptual model—a collection of assumptions about the components and the structure of the system, plus hypotheses about the values of model input parameters.
- The third step is the implementation of an operational model, usually by using simulation software and incorporating the assumptions of the conceptual model into the world-view and concepts of the simulation software.
- In actuality, model building is not a linear process with three steps.

## Model Building Process

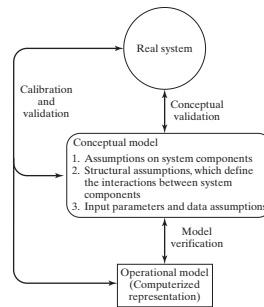


Figure : Model Building, Verification and Validation

## Verification

The purpose of model verification is to assure that the conceptual model is reflected accurately in the operational model. Many common-sense suggestions can be given for use in the verification process:

- ① Have the operational model checked by someone other than its developer, preferably an expert in the simulation software being used?
- ② Make a flow diagram that 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.
- ③ Closely examine the model output for reasonableness under a variety of settings of the input parameters.

## Verification

- 4 Have the operational model 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.
- 8 Graphical interfaces are recommended for accomplishing verification and validation.

## Calibration and Validation of Models

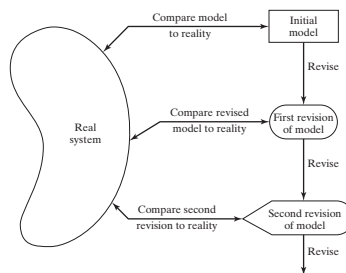


Figure : Iterative Process of Calibrating a Model

## Calibration and Validation of Models

We can follow a three-step approach for validation.

- ❶ Build a model that has high face validity.
- ❷ Validate model assumptions.
- ❸ Compare the model input-output transformations to corresponding input-output transformations for the real system.

## 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. The potential users of a model should be involved in model construction from its conceptualization and to its implementation, to ensure that a high degree of realism is built into the model through reasonable assumptions regarding system structure and through reliable data. These people can also evaluate model output for reasonableness and can aid in identifying model deficiencies. Sensitivity analysis can also be used to check a model's face validity. The model user is asked whether the model behaves in the expected way when one or more input variables is changed.

## 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. Data assumptions should be based on the collection of reliable data and correct statistical analysis of the data. The analysis consists of three steps:

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

## Validating I-O Transformations

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. A necessary condition for the validation of input-output transformations is that some version of the system under study exist, so that system data under at least one set of input conditions can be collected to compare to model 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.

## Example

A bank 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. Service times and inter-arrival times were collected for 90 customers who arrived between 11.00 AM and 1.00 PM on a Friday as it is considered as a representative time period of a typical rush hour. Data analysis 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 mean 1.1 minutes and standard deviation 0.2 minutes.

- ❶ inter-arrival times, Poisson with rate  $\lambda = 45$  per hour.
- ❷ service times, normal with  $\mu = 1.1$  and  $\sigma = 0.2$ .

## Example

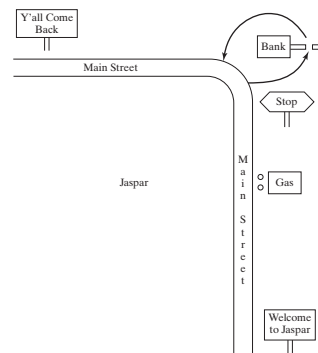


Figure : Drive-In Window

## Example

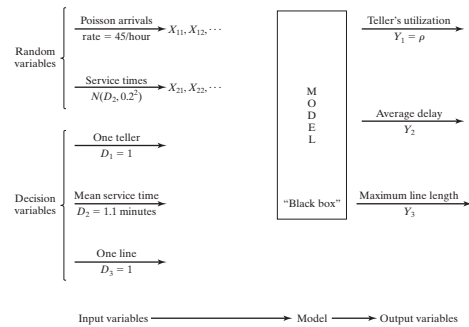


Figure : Model Input-Output Transformation

## I-O Variables

Input Variables	Output Variables
<b>X: Other Variables</b>	<b>Variables of Primary Interest</b>
Poisson Arrivals $X_{11}, X_{12}, \dots$	$Y_1$ : teller's utilization
Service Times $X_{21}, X_{22}, \dots$	$Y_2$ : average delay
	$Y_3$ : max line length
<b>D: Decision Variables</b>	<b>Variables of Secondary Interest</b>
$D_1 = 1$ (one teller)	$Y_4$ : observed arrival rate
$D_2 = 1.1$ (mean service time)	$Y_5$ : average service rate
$D_3 = 1$ (one line)	$Y_6$ : sample std. dev. of service times
	$Y_7$ : average length of waiting line

Table : Input and Output Variables for the Model



## Example

Replication	$Y_2$ (minutes)	$Y_4$ (arrivals / hour)	$Y_5$ (minutes)
1	2.79	51.0	1.07
2	1.12	40.0	1.12
3	2.24	45.5	1.06
4	3.45	50.5	1.10
5	3.13	53.0	1.09
6	2.38	49.0	1.07
Mean	2.51		
Std. Dev.	0.82		

Table : Results of Six Replications of the Initial Model

## Example

We want to perform a validation test where we compare the system response time,  $Z_2 = 4.3$  minutes, to the model response time,  $Y_2$ . Formally, a statistical test as follows is conducted:

- $H_0 : E(Y_2) = 4.3$  minutes
- $H_1 : E(Y_2) \neq 4.3$  minutes

## Example

If the null hypothesis is not rejected, then, on the basis of this test, there is no reason to consider the model invalid. If the null hypothesis is rejected, the current version of the model is rejected, and the modeler is forced to seek ways to improve the model. We choose a level of significance and a sample size. We here choose  $\alpha = 0.05$  and  $n = 6$ .

## Example

Compute the sample mean,  $\bar{Y}_2$ , and standard deviation,  $S$ , over  $n$  replications.

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

$$S = \sqrt{\frac{\sum_{i=1}^n (Y_{2i} - \bar{Y}_2)^2}{n-1}} = 0.82 \text{ minutes}$$

where  $Y_{2i}, i = 1, \dots, 6$ , are shown in the table of results of six replications.

## Example

We now read the critical  $t$  value from Table A.5 in the text. For a one sided test, we use  $t_{\alpha, n-1}$  and for a two-sided test, as it is here, we use  $t_{\alpha/2, n-1}$ . Here,  $t_{0.025, 5} = 2.571$  for a two-sided test. We compute the test statistic as

$$t_0 = \frac{\bar{Y}_2 - \mu_0}{S/\sqrt{n}} = \frac{2.51 - 4.3}{0.82/\sqrt{6}} = -5.34 \quad (1)$$

For a two-sided test, we do not reject  $H_0$  if

$$-t_{\alpha/2, n-1} < t_0 < +t_{\alpha/2, n-1}$$

Here, since  $t_0 = 5.34 > t_{0.025, 5} = 2.571$ , we reject  $H_0$ , and conclude that the model is not adequate in its prediction of average customer delay.

## Example

Now that the model of the bank has been found lacking, what should do modeler do? Upon further investigating, 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.

## Example

To correct the 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 run, yielding the results in the following table.

Replication	$Y_2$ (minutes)	$Y_4$ (arrivals / hour)	$Y_5$ (minutes)
1	5.37	51.0	1.07
2	1.98	40.0	1.12
3	5.29	45.5	1.06
4	3.82	50.5	1.09
5	6.74	53.0	1.08
6	5.49	49.0	1.08
Mean	4.78		
Std. Dev.	1.66		

Table : Results of Six Replications of the Revised Model

## Example

A hypothesis test as before was conducted. We have the following procedure:

- Choose  $\alpha = 0.05$ ,  $n = 6$ .
- Compute  $\bar{Y}_2 = 4.78$  minutes and  $S = 1.66$  minutes.
- Critical  $t$  value is  $t_{0.25,5} = 2.571$
- Compute  $t_0 = (\bar{Y}_2 - \mu_0)/(S/\sqrt{n}) = 0.710$ .
- Since  $t_0 < t_{0.025,5}$ , we do not reject the null hypothesis.

## Example

We continue to assume that there is a known output performance measure for the existing system, denoted by  $\mu_0$ , and an unknown performance measure of the simulation,  $\mu$ , that we hope is close. The hypothesis-testing formulation tested whether  $\mu = \mu_0$ ; the confidence-interval (CI) formulation tries to bound the difference, i.e.,  $|\mu - \mu_0| \leq \epsilon$ .

Specifically, if  $Y$  is the simulation output and  $\mu = E(Y)$ , then, we execute the simulation and form a confidence interval for  $\mu$ , such as  $\bar{Y} \pm t_{\alpha/2, n-1} S / \sqrt{n}$ . The determination of whether to accept the model as valid depends on the best-case and worst-case error implied by the CI.

## Example

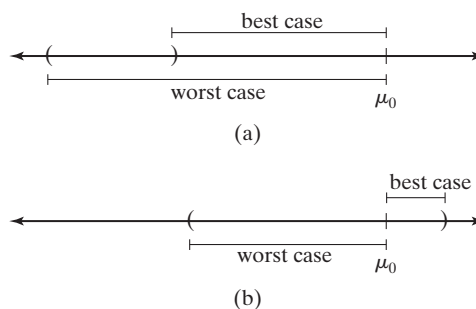


Figure : Validation of the Input-Output Transformation

## Example

If the CI does not contain  $\mu_0$ .

- ❶ If the best-case error is greater than  $\epsilon$ , then, the difference in performance is large enough, even in the best case, to indicate that we need to refine the model.
- ❷ If the worst-case error is less than or equal to  $\epsilon$ , then, we can accept the simulation model as close enough to be considered valid.
- ❸ If the best-case error is less than or equal to  $\epsilon$ , but the worst-case error is greater than  $\epsilon$ , then, additional replications are necessary to shrink the CI until a conclusion can be reached.

## Example

If the CI does contain  $\mu_0$ .

- ❶ If either the best-case or worst-case error is greater than  $\epsilon$ , then, additional replications are necessary to shrink the CI until a conclusion can be reached.
- ❷ If the worst-case error is less than or equal to  $\epsilon$ , then, we can accept the simulation model as close enough to be considered valid.

## I-O Validation using Historical Data

An alternative to generating input data is to use the actual historical record to drive the simulation model and then to compare model output with the system data. To implement this technique  $A_1, A_2, \dots$  and  $S_1, S_2, \dots$  would have to be entered into the model into arrays or stored in a file to be read as the need arose. This event scheduling without random-number generation could be implemented quite easily in a general-purpose programming language or most simulation languages by using arrays to store the data or reading the data from a file.

## Example

### Example 10.3: The Candy Factory

A simulation model was developed, and a validation effort using historical inputs was conducted. The system input data,  $T_{ij}$  and  $D_{ij}$ , were fed into the model and used as run times and random downtimes. Model response variables ( $Y_i, i = 1, 2, 3$ ) were collected for comparison to the corresponding system response variables ( $Z_i, i = 1, 2, 3$ ).

## Example

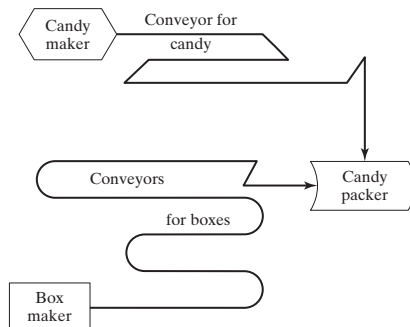


Figure : Production Line at the Candy Factory

## Example

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 the following table. A simple display such as this table can be quite effective in convincing people of a model's validity.



## Example

Response, $i$	System, $Z_i$	Model, $Y_i$
1. production level	897,208	883,150
2. # of operator interventions	3	3
3. time of occurrence	7:22, 8:41, 10:10	7:24, 8:42, 10:14

Table : Validation of Candy Factory Model

## Example

$$H_0 : \mu_d = 0$$

$$H_1 : \mu_d \neq 0$$

Set	$Z_{ij}$	$W_{ij}$	$d_j$	$(d_j - \bar{d})^2$
1	$Z_{i1}$	$W_{i1}$	$d_1 = Z_{i1} - W_{i1}$	$(d_j - \bar{d})^2$
1	$Z_{i1}$	$W_{i1}$	$d_1 = Z_{i1} - W_{i1}$	$(d_j - \bar{d})^2$
...	...	...	...	...
K	$Z_{iK}$	$W_{iK}$	$d_K = Z_{iK} - W_{iK}$	$(d_K - \bar{d})^2$
$\bar{d} = \frac{1}{K} \sum_{j=1}^K d_j \quad S_d^2 = \frac{1}{K-1} \sum_{j=1}^K (d_j - \bar{d})^2$				

Table : Comparison of System and Model Output Measures

## Example

The proper test is a paired  $t$  test ( $Z_{i1}$  is paired with  $W_{i1}$ , each having been 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 given formulas. Then, compute the  $t$  statistic as

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

(with  $\mu_d = 0$ ), and get the critical value  $t_{\alpha/2, K-1}$  from table A.5 in the text. If  $-t_{\alpha/2, K-1} \leq t_0 \leq +t_{\alpha/2, K-1}$ , do not reject  $H_0$ , and hence, conclude that this test provides no evidence of model inadequacy.

## Example

The Candy Factory Example 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 in the text. A paired  $t$  test was conducted to test where  $Z_1$  is the system production level and  $W_1$  is the model prediction of the system production level. The hypotheses are

$$H_0 : \mu_d = E(Z_1) - E(W_1) = 0$$

$$H_1 : \mu_d \neq 0$$

## Example

For  $\alpha = 0.05$ , we have

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

Hence, since  $-t_{0.025,4} = -2.78 \leq t_0 = 1.37 \leq +t_{\alpha/2,K-1} = +2.78$ , the null hypothesis is not rejected.

## I-O 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 shuffled and given to the engineer to be asked to decide which are fake and which are real. This type of validation test is commonly called a Turing test.

## Summary

- Reading HW: Chapter 10.
- Chapter 10 Exercises.