Chapter 10

Verification and Validation of Simulation Models
Contents

- Model-Building, Verification, and Validation
- Verification of Simulation Models
- Calibration and Validation
Purpose & Overview

• The goal of the validation process is:
  • To produce a model that represents true behavior closely enough for decision-making purposes
  • To increase the model’s credibility to an acceptable level

• 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

• Most methods are informal subjective comparisons while a few are formal statistical procedures
Modeling-Building, Verification & Validation
Modeling-Building, Verification & Validation

- Steps in Model-Building
  - Observing the real system and the interactions among their various components and of collecting data on their behavior
  - Construction of a conceptual model
  - Implementation of an operational model
Verification

- Purpose: ensure the conceptual model is reflected accurately in the computerized representation.
- Many common-sense suggestions, for example:
  - Have someone else check the model.
  - Make a flow diagram that includes each logically possible action a system can take when an event occurs.
  - Closely examine the model output for reasonableness under a variety of input parameter settings.
  - Print the input parameters at the end of the simulation, make sure they have not been changed inadvertently.
  - Make the operational model as self-documenting as possible.
  - If the operational model is animated, verify that what is seen in the animation imitates the actual system.
  - Use the debugger.
  - If possible use a graphical representation of the model.
Examination of Model Output for Reasonableness

- Two statistics that give a quick indication of model reasonableness are **current contents** and **total counts**
  - Current content: The number of items in each component of the system at a given time.
  - Total counts: Total number of items that have entered each component of the system by a given time.
- Compute certain long-run measures of performance, e.g. compute the long-run server utilization and compare to simulation results.
Examination of Model Output for Reasonableness

- A model of a complex network of queues consisting of many service centers.
  - If the current content grows in a more or less linear fashion as the simulation run time increases, it is likely that a queue is unstable.
  - If the total count for some subsystem is zero, indicates no items entered that subsystem, a highly suspect occurrence.
  - If the total and current count are equal to one, can indicate that an entity has captured a resource but never freed that resource.

![Diagram of a complex network of queues](image-url)
Other Important Tools

• Documentation
  • A means of clarifying the logic of a model and verifying its completeness.
  • Comment the operational model, definition of all variables and parameters.

• Use of a trace
  • A detailed printout of the state of the simulation model over time.
  • Can be very labor intensive if the programming language does not support statistic collection.
  • Labor can be reduced by a centralized tracing mechanism
  • In object-oriented simulation framework, trace support can be integrated into class hierarchy. New classes need only to add little for the trace support.
Trace: Example

- Simple queue from Chapter 2
- Trace over a time interval [0, 16]
- Allows the test of the results by pen-and-paper method

**Definition of Variables:**
- **CLOCK** = Simulation clock
- **EVTYP** = Event type (Start, Arrival, Departure, 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:**

<table>
<thead>
<tr>
<th>CLOCK</th>
<th>EVTYP</th>
<th>NCUST</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Start</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Arrival</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Depart</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>Arrival</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>Arrival</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>Depart</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

There is a customer, but the status is 0
Calibration and Validation
Calibration and Validation

- Validation: the overall process of comparing the model and its behavior to the real system.
- Calibration: the iterative process of comparing the model to the real system and making adjustments.
- Comparison of the model to real system
  - Subjective tests
    - People who are knowledgeable about the system
  - Objective tests
    - Requires data on the real system’s behavior and the output of the model
Calibration and Validation

• 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

• 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.

• Three-step approach for validation:
  • Build a model that has high face validity.
  • Validate model assumptions.
  • Compare the model input-output transformations with the real system’s data.
High Face Validity

- Ensure a high degree of realism:
  - Potential users should be involved in model construction from its conceptualization to its implementation.
- Sensitivity analysis can also be used to check a model’s face validity.
  - Example: In most queueing systems, if the arrival rate of customers were to increase, it would be expected that server utilization, queue length and delays would tend to increase.
  - For large-scale simulation models, there are many input variables and thus possibly many sensitivity tests.
    - Sometimes not possible to perform all of these tests, select the most critical ones.
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
    - Customer waiting in one line versus many lines
    - Customers are served according 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
Validate Input-Output Transformation

- **Goal:** Validate the model’s ability to predict future behavior
  - The only objective test of the model.
  - The structure of the model should be accurate enough to make good predictions for the range of input data sets of interest.
- **One possible approach:** use historical data that have been reserved for validation purposes only.
- **Criteria:** use the main responses of interest.
Bank Example

- Example: One drive-in window serviced by one teller, only one or two transactions are allowed.
- Data collection: 90 customers during 11 am to 1 pm.
  - Observed service times \( \{S_i, i = 1, 2, \ldots, 90\} \).
  - Observed interarrival times \( \{A_i, i = 1, 2, \ldots, 90\} \).
- Data analysis let to the conclusion that:
  - Interarrival times: exponentially distributed with rate \( \lambda = 45/\text{hour} \)
  - Service times: \( N(1.1, 0.2^2) \)

\[ 
\text{Input variables} \]

Prof. Dr. Mesut Güneş • Ch. 10 Verification and Validation of Simulation Models
Bank Example: The Black Box

- A model was developed in close consultation with bank management and employees
- Model assumptions were validated
- Resulting model is now viewed as a “black box”:

Input Variables

- Poisson arrivals
  \( \lambda = 45/hr: X_{11}, X_{12}, \ldots \)
- Services times
  \( N(D_2, 0.22): X_{21}, X_{22}, \ldots \)

\( D_1 = 1 \) (one teller)
\( D_2 = 1.1 \) min
(mean service time)
\( D_3 = 1 \) (one line)

Model “black box”

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

Model Output Variables, \( Y \)

Primary interest:
- \( Y_1 \) = teller’s utilization
- \( Y_2 \) = average delay
- \( Y_3 \) = maximum line length

Secondary interest:
- \( Y_4 \) = observed arrival rate
- \( Y_5 \) = average service time
- \( Y_6 \) = sample std. dev. of service times
- \( Y_7 \) = average length of time
Comparison with Real System Data

- Real system data are necessary for validation.
  - System responses should have been collected during the same time period (from 11am to 1pm on the same day.)
- Compare average delay from the model $Y_2$ with actual delay $Z_2$:
  - Average delay observed, $Z_2 = 4.3$ minutes, consider this to be the true mean value $\mu_0 = 4.3$.
  - When the model is run with generated random variates $X_{1n}$ and $X_{2n}$, $Y_2$ should be close to $Z_2$. 
Comparison with Real System Data

- Six statistically independent replications of the model, each of 2-hour duration, are run.

<table>
<thead>
<tr>
<th>Replication</th>
<th>$Y_4$ Arrivals/Hour</th>
<th>$Y_5$ Service Time [Minutes]</th>
<th>$Y_2$ Average Delay [Minutes]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>51.0</td>
<td>1.07</td>
<td>2.79</td>
</tr>
<tr>
<td>2</td>
<td>40.0</td>
<td>1.12</td>
<td>1.12</td>
</tr>
<tr>
<td>3</td>
<td>45.5</td>
<td>1.06</td>
<td>2.24</td>
</tr>
<tr>
<td>4</td>
<td>50.5</td>
<td>1.10</td>
<td>3.45</td>
</tr>
<tr>
<td>5</td>
<td>53.0</td>
<td>1.09</td>
<td>3.13</td>
</tr>
<tr>
<td>6</td>
<td>49.0</td>
<td>1.07</td>
<td>2.38</td>
</tr>
</tbody>
</table>

- Sample mean [Delay] 2.51
- Standard deviation [Delay] 0.82
Hypothesis Testing

- Compare the average delay from the model $Y_2$ with the actual delay $Z_2$

- Null hypothesis testing: evaluate whether the simulation and the real system are the same (w.r.t. output measures):

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

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

- If $H_0$ is not rejected, then, 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 needs to improve the model
Hypothesis Testing

- Conduct the $t$ test:
  - Chose level of significance ($\alpha = 0.05$) and sample size ($n = 6$).
  - Compute the sample mean and sample standard deviation over the $n$ replications:
    
    $\overline{Y}_2 = \frac{1}{n} \sum_{i=1}^{n} Y_{2i} = 2.51$ minutes
    
    $S = \sqrt{\frac{\sum_{i=1}^{n} (Y_{2i} - \overline{Y}_2)^2}{n-1}} = 0.82$ minutes

- Compute test statistics:

  $|t_0| = \left| \frac{\overline{Y}_2 - \mu_0}{S / \sqrt{n}} \right| = \left| \frac{2.51 - 4.3}{0.82 / \sqrt{6}} \right| = 5.34 > t_{critical} = 2.571$ (for a 2-sided test)

- Hence, reject $H_0$.
  - Conclude that the model is inadequate.
  - Check: the assumptions justifying a $t$ test, that the observations ($Y_{2i}$) are normally and independently distributed.
Hypothesis Testing

• Similarly, compare the model output with the observed output for other measures:

\[ Y_4 \leftrightarrow Z_4, \ Y_5 \leftrightarrow Z_5, \ and \ Y_6 \leftrightarrow Z_6 \]
Type II Error

• For validation, the power of the test is:
  • Probability(detecting an invalid model) = 1 − β
  • β = P(Type II error) = P(failing to reject $H_0 \mid H_1$ is true)
  • Consider failure to reject $H_0$ as a strong conclusion, the modeler would want β to be small.
  • Value of β depends on:
    • Sample size, $n$
    • The true difference, $\delta$, between $E(Y)$ and $\mu$:
      $$\delta = \frac{|E(Y) - \mu|}{\sigma}$$

• In general, the best approach to control β error is:
  • Specify the critical difference, $\delta$.
  • Choose a sample size, $n$, by making use of the operating characteristics curve (OC curve).
Type II Error

- Operating characteristics curve (OC curve).
- Graphs of the probability of a Type II Error $\beta(\delta)$ versus $\delta$ for a given sample size $n$.

For the same error probability with smaller difference the required sample size increases!
Type I and II Error

- **Type I error ($\alpha$):**
  - Error of rejecting a valid model.
  - Controlled by specifying a small level of significance $\alpha$.
- **Type II error ($\beta$):**
  - Error of accepting a model as valid when it is invalid.
  - Controlled by specifying critical difference and find the $n$.
- For a fixed sample size $n$, increasing $\alpha$ will decrease $\beta$.

<table>
<thead>
<tr>
<th>Statistical Terminology</th>
<th>Modeling Terminology</th>
<th>Associated Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type I: rejecting $H_0$ when $H_0$ is true</td>
<td>Rejecting a valid model</td>
<td>$\alpha$</td>
</tr>
<tr>
<td>Type II: failure to reject $H_0$ when $H_1$ is true</td>
<td>Failure to reject an invalid model</td>
<td>$\beta$</td>
</tr>
</tbody>
</table>
Confidence Interval Testing

- Confidence interval testing: evaluate whether the simulation and the real system performance measures are close enough.
- If $Y$ is the simulation output, and $\mu = E(Y)$
- The confidence interval (CI) for $\mu$ is:

$$
\left[ \bar{Y} - t_{\frac{\alpha}{2}, n-1} \frac{S}{\sqrt{n}}, \bar{Y} + t_{\frac{\alpha}{2}, n-1} \frac{S}{\sqrt{n}} \right]
$$
Confidence Interval Testing

- Validating the model:
  - Suppose the CI does not contain $\mu_0$:
    - If the best-case error is $> \varepsilon$, model needs to be refined.
    - If the worst-case error is $\leq \varepsilon$, accept the model.
    - If best-case error is $\leq \varepsilon$, additional replications are necessary.
  - Suppose the CI contains $\mu_0$:
    - If either the best-case or worst-case error is $> \varepsilon$, additional replications are necessary.
    - If the worst-case error is $\leq \varepsilon$, accept the model.

$\varepsilon$ is a difference value chosen by the analyst, that is small enough to allow valid decisions to be based on simulations!

Prof. Dr. Mesut Güneş • Ch. 10 Verification and Validation of Simulation Models
Confidence Interval Testing

- Bank example: $\mu_0 = 4.3$, and “close enough” is $\varepsilon = 1$ minute of expected customer delay.
  - A 95% confidence interval, based on the 6 replications is $[1.65, 3.37]$ because:

$$\bar{Y} \pm t_{0.025, 5} \frac{S}{\sqrt{n}}$$

$$2.51 \pm 2.571 \frac{0.82}{\sqrt{6}}$$

- $\mu_0 = 4.3$ falls outside the confidence interval,
  - the best case $|3.37 - 4.3| = 0.93 < 1$, but
  - the worst case $|1.65 - 4.3| = 2.65 > 1$

- Additional replications are needed to reach a decision.
Using Historical Output Data

- An alternative to generating input data:
  - Use the actual historical record.
  - Drive the simulation model with the historical record and then compare model output to system data.
  - In the bank example, use the recorded interarrival and service times for the customers \( \{A_n, S_n, n = 1,2,\ldots\} \).

- Procedure and validation process: similar to the approach used for system generated input data.
Using a Turing Test

- Use in addition to statistical test, or when no statistical test is readily applicable.

**Turing Test**
Described by Alan Turing in 1950. A human judge is involved in a natural language conversation with a human and a machine. If the judge cannot reliably tell which of the partners is the machine, then the machine has passed the test.

- Utilize persons’ knowledge about the system.
- For example:
  - Present 10 system performance reports to a manager of the system. Five of them are from the real system and the rest are “fake” reports based on simulation output data.
  - If the person identifies a substantial number of the fake reports, interview the person to get information for model improvement.
  - If the person cannot distinguish between fake and real reports with consistency, conclude that the test gives no evidence of model inadequacy.
Summary

- Model validation is essential:
  - Model verification
  - Calibration and validation
  - Conceptual validation
- Best to compare system data to model data, and make comparison using a wide variety of techniques.
- Some techniques that we covered:
  - Insure high face validity by consulting knowledgeable persons.
  - Conduct simple statistical tests on assumed distributional forms.
  - Conduct a Turing test.
  - Compare model output to system output by statistical tests.