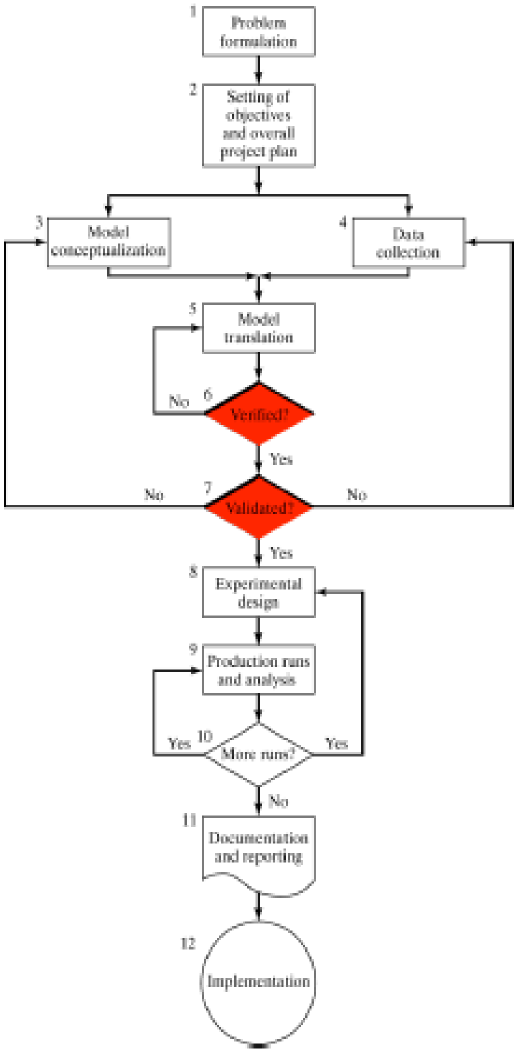
Verification and Validation of Simulation Models

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



Verification and Validation of Simulation Models

- Verification: concerned with building the ***model right***. It is utilized in the comparison of the conceptual model to the computer representation that implements that conception.

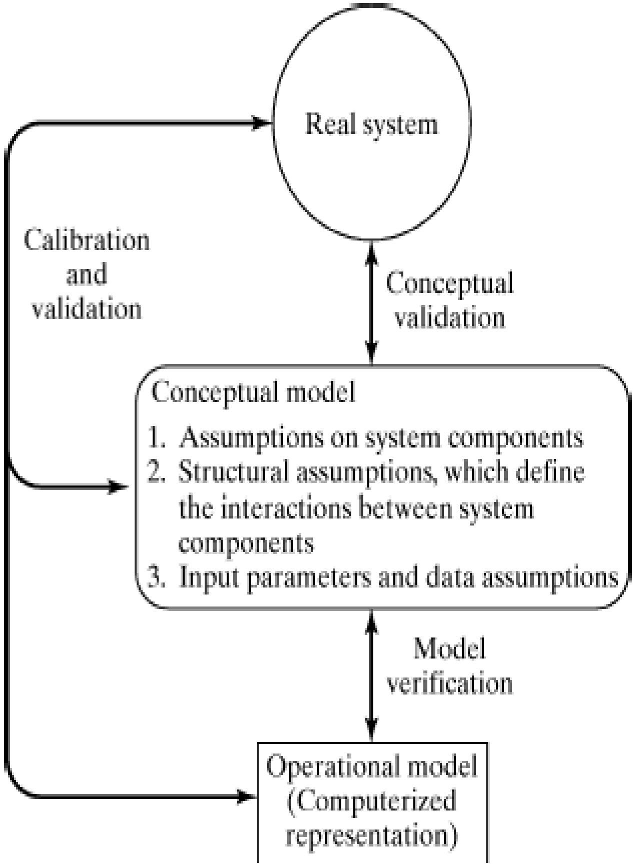
- It asks the questions: Is the model implemented correctly in the computer?
- Are the input parameters and logical structure of the model correctly represented?

Verification and Validation of Simulation Models (cont.)

- Validation: 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.

Modeling-Building, Verification & Validation

- Steps in Model-Building
  - Real system
    - Observe the real system
    - Interactions among the components
    - Collecting data on the behavior
  - Conceptual model
    Construction of a conceptual model
  - Simulation program
    Implementation of an operational model


Real system
Calibration and validation
Conceptual validation
Conceptual model
1. Assumptions on system components
2. Structural assumptions, which define the interactions between system components
3. Input parameters and data assumptions
Model verification
Operational model (Computerized representation)

Verification of Simulation Models

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

1. Have the code checked by someone other than the programmer.
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.
3. Closely examine the model output for reasonableness under a variety of settings of the input parameters. Have the code print out a wide variety of output statistics.
4. Have the computerized model print the input parameters at the end of the simulation, to be sure that these parameter values have not been changed inadvertently.

Verification of Simulation Models

5. Make the computer code 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.

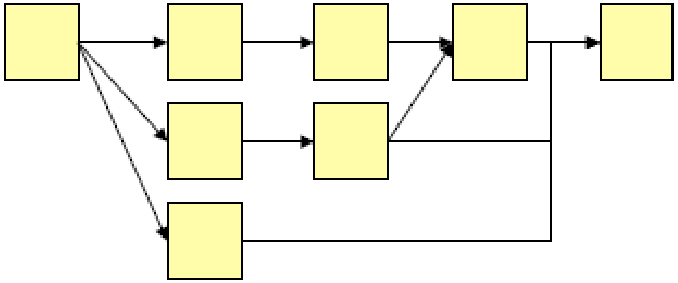
These suggestions are basically the same ones any programmer would follow when debugging a computer program.

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.



Documentation

- Documentation
  - A means of clarifying the logic of a model and verifying its completeness.
  - Comment the operational model
    - definition of all variables (default values?)
    - definition of all constants (default values?)
    - functions and parameters
    - relationship of objects
    - etc.

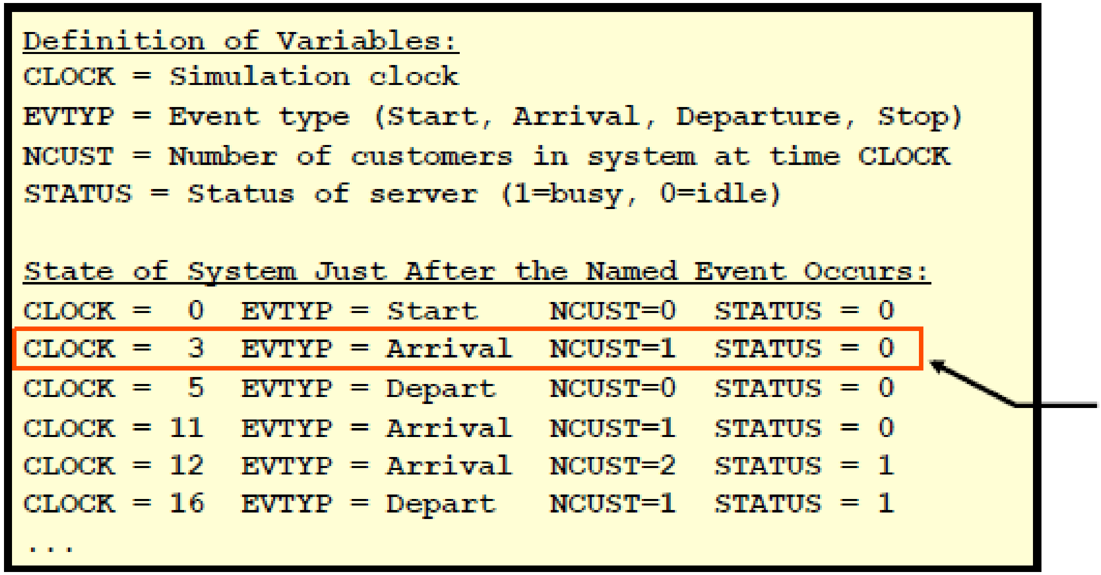
- Default values should be explained!

Trace

- A trace is 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

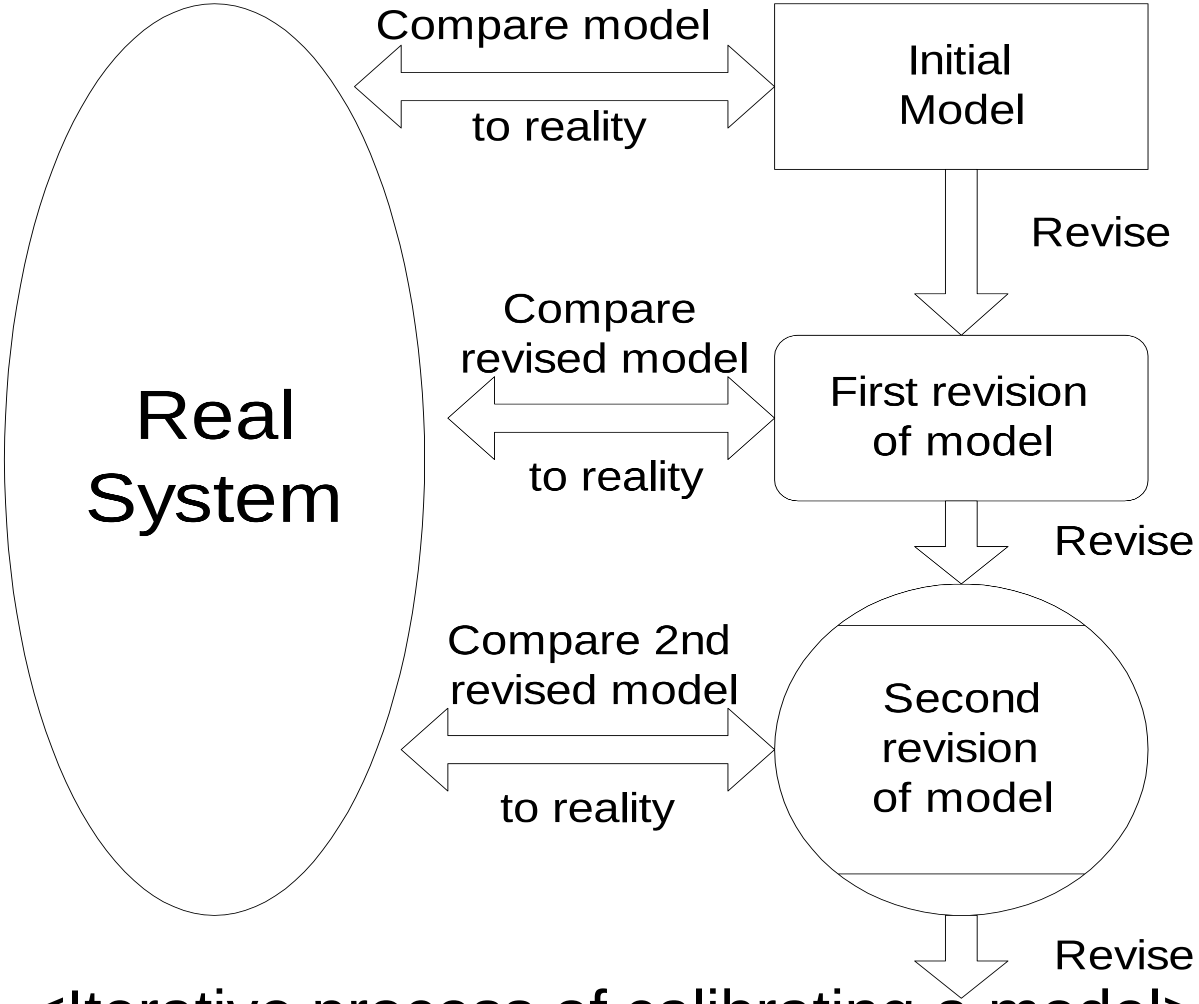


```
Definition of Variables:
CLOCK = Simulation clock
EVTYP = Event type (Start, Arrival, Departure, Stop)
NCUST = Number of customers in system at time CLOCK
STATUS = Status of server (1=busy, 0=idle)

State of System Just After the Named Event Occurs:
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
...
```

There is a customer, but the status is 0

Calibration and Validation of Models



<Iterative process of calibrating a model>

- 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

Validation of Simulation Models

As an aid in the validation process, Naylor and Finger 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.

Validation: 1. High Face Validity

- Ensure a high degree of realism:
  - Potential users should be involved in model construction from its conceptualization to its implementation.

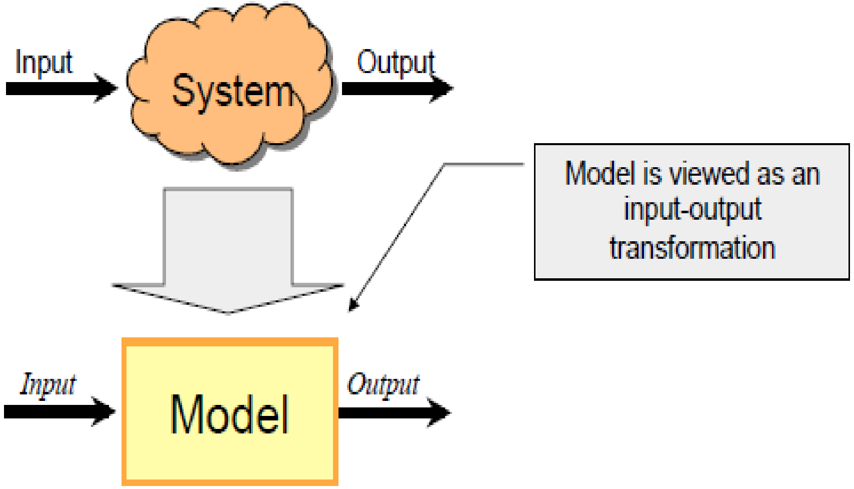
- Sensitivity analysis can also be used to check a model's face validity.
  - Example: In most queueing systems, if the arrival rate of customers were to increase, it would be expected that server utilization, queue length and delays would tend to increase.
  - For large-scale simulation models, there are many input variables and thus possibly many sensitivity tests.
    - Sometimes not possible to perform all of theses tests, select the most critical ones.

Validation: 2. 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

Validation:
3. 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.



Validation of Model Assumptions

The analysis of input data from a random sample 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.

Bank Example

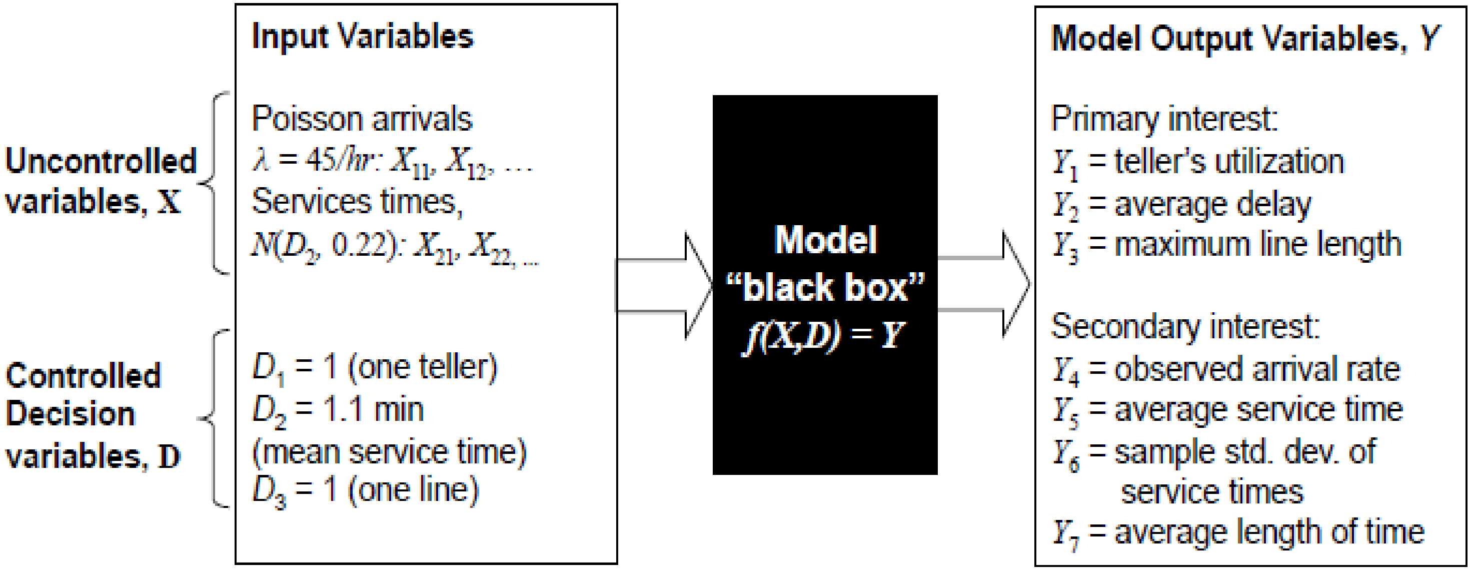
Example: One drive-in window serviced by one teller, only one or two transactions are allowed.

- Data collection: 90 customers during 11am to 1pm
  - 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 λ = 45/hour
  - Service times: $N(1.1, 0.2^2)$

Input variables

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":



Uncontrolled variables, X

Controlled Decision variables, D

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

Bank Example: Comparison with Real System Data

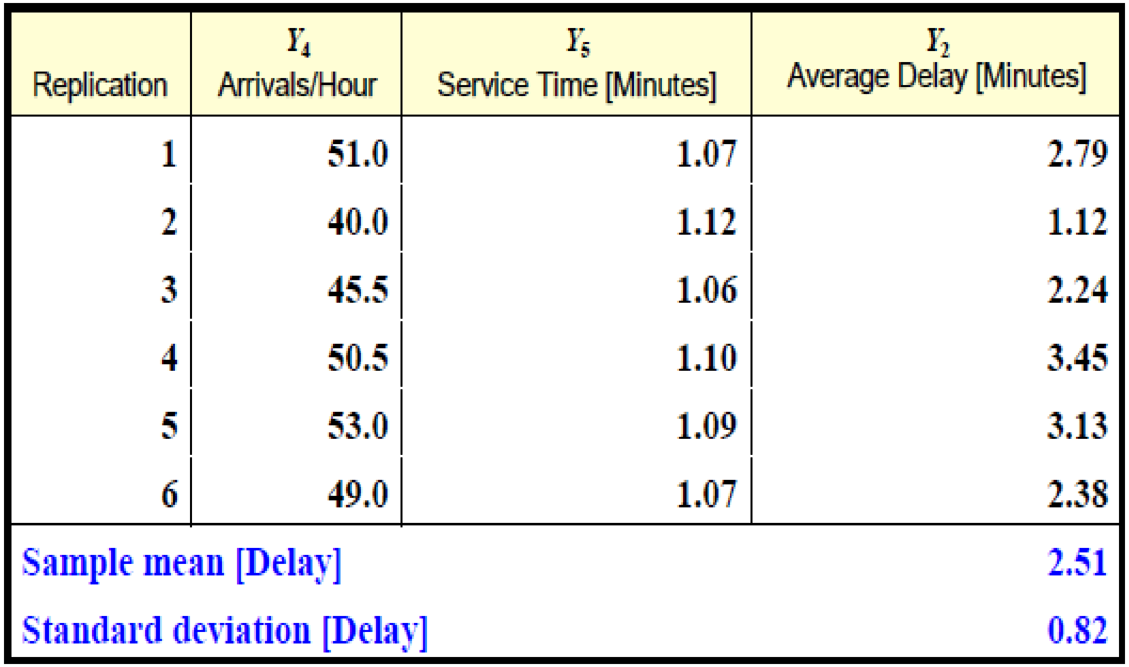
- Real system data are necessary for validation.
  - System responses should have been collected during the same time period (from 11am to 1pm on the same day.)

- 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

Bank Example: Comparison with Real System Data

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



Replication	Y_4 Arrivals/Hour	Y_5 Service Time [Minutes]	Y_2 Average Delay [Minutes]
1	51.0	1.07	2.79
2	40.0	1.12	1.12
3	45.5	1.06	2.24
4	50.5	1.10	3.45
5	53.0	1.09	3.13
6	49.0	1.07	2.38
Sample mean [Delay]			**2.51**
Standard deviation [Delay]			**0.82**

Bank Example: 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

Bank Example: 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:

$$\bar{Y}_2 = \frac{1}{n}\sum_{i=1}^{n} Y_{2i} = 2.51 \text{ minutes} \qquad S = \sqrt{\frac{\sum_{i=1}^{n}(Y_{2i} - \bar{Y}_2)^2}{n-1}} = 0.82 \text{ minutes}$$

  - Compute test statistics:

$$|t_0| = \left|\frac{\bar{Y}_2 - \mu_0}{S/\sqrt{n}}\right| = \left|\frac{2.51 - 4.3}{0.82/\sqrt{6}}\right| = 5.34 > t_{0.025,5} = 2.571 \quad \text{(for a 2-sided test)}$$

  - 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.

Bank Example: Hypothesis Testing

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

$$Y_4 \leftrightarrow Z_4$$

$$Y_5 \leftrightarrow Z_5$$

$$Y_6 \leftrightarrow Z_6$$

Power of a test

- For validation:

The power of a test is the probability of detecting an invalid model.

$$\begin{aligned} Power &= 1 - P(\text{failing to reject } H_0 \mid H_1 \text{ is true}) \\ &= 1 - P(\text{Type II error}) \\ &= 1 - \beta \end{aligned}$$

- Consider failure to reject H_0 as a strong conclusion, the modeler would want β to be small.

Power of a test

- Value of β depends on:
  - Sample size n
  - The true difference, δ, between $E(Y)$ and μ

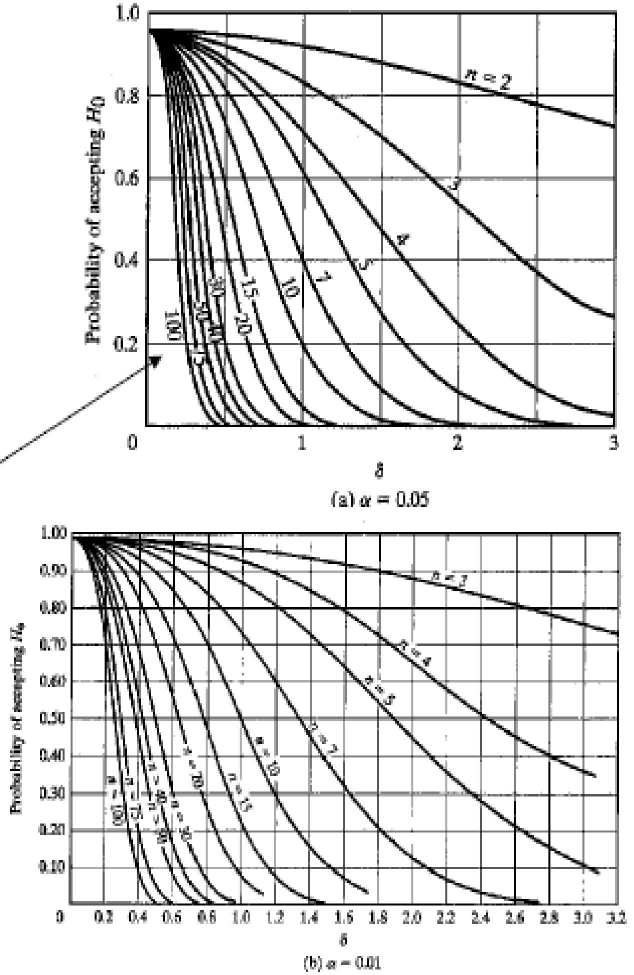
$$\delta = \frac{|E(Y) - \mu|}{\sigma}$$

- In general, the best approach to control β is:
  - Specify the critical difference, δ.
  - Choose a sample size, n, by making use of the operating characteristics curve (OC curve).

Power of a test

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

For the same error probability with smaller difference the required sample size increases!

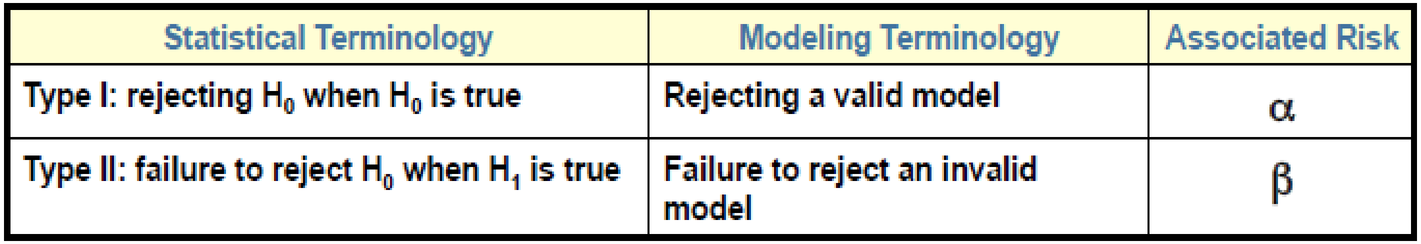


(a) $\alpha = 0.05$

(b) $\alpha = 0.01$

Power of a test

- Type I error (α):
  - Error of rejecting a valid model.
  - Controlled by specifying a small level of significance α.
- Type II error (β):
  - 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 α will decrease β.



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

Confidence Interval Testing

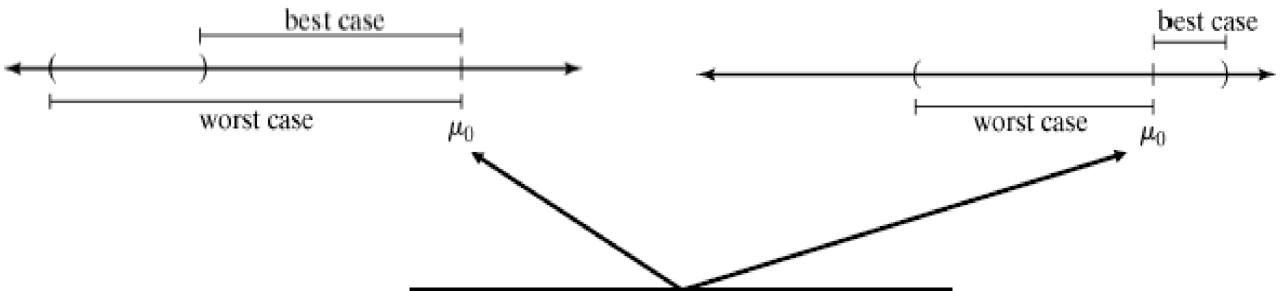
- Confidence interval testing: evaluate whether the simulation and the real system performance measures are close enough.
- If Y is the simulation output and $\mu = E(Y)$
- The confidence interval (CI) for μ is:

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

Confidence Interval Testing

- CI does not contain μ_0:
  - If the best-case error is > ε, model needs to be refined.
  - If the worst-case error is ≤ ε, accept the model.
  - If best-case error is ≤ ε, additional replications are necessary.
- CI contains μ_0:
  - If either the best-case or worst-case error is > ε, additional replications are necessary.
  - If the worst-case error is ≤ ε, accept the model.

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



μ_0 is the unknown true value

Confidence Interval Testing

- Bank example: $\mu_0 = 4.3$, and "close enough" is ε = 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.

Other approaches

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 $\{A_n, S_n, n = 1,2,\ldots\}$.

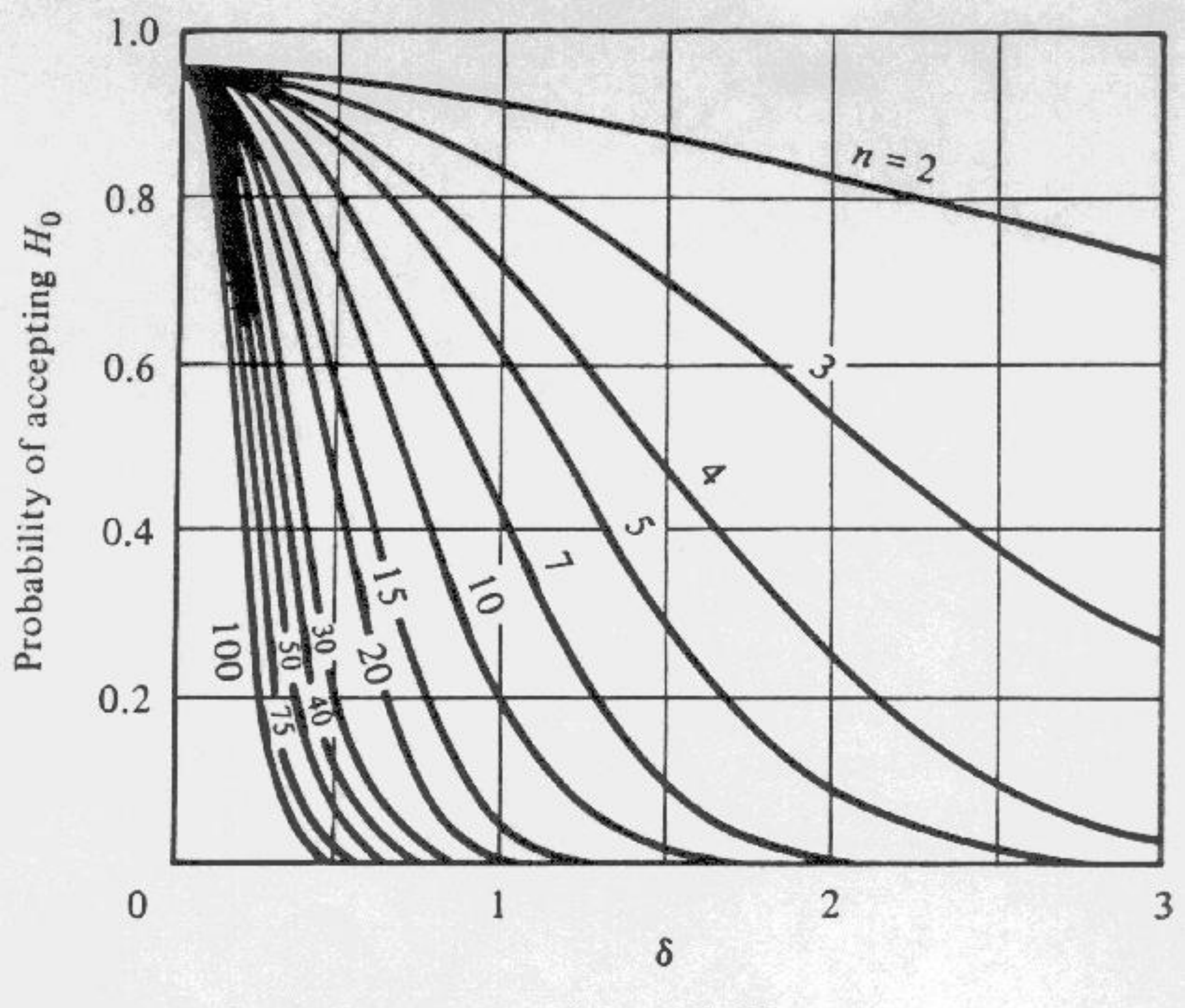
- Procedure and validation process: similar to the approach used for system generated input data.

Using a Turing Test

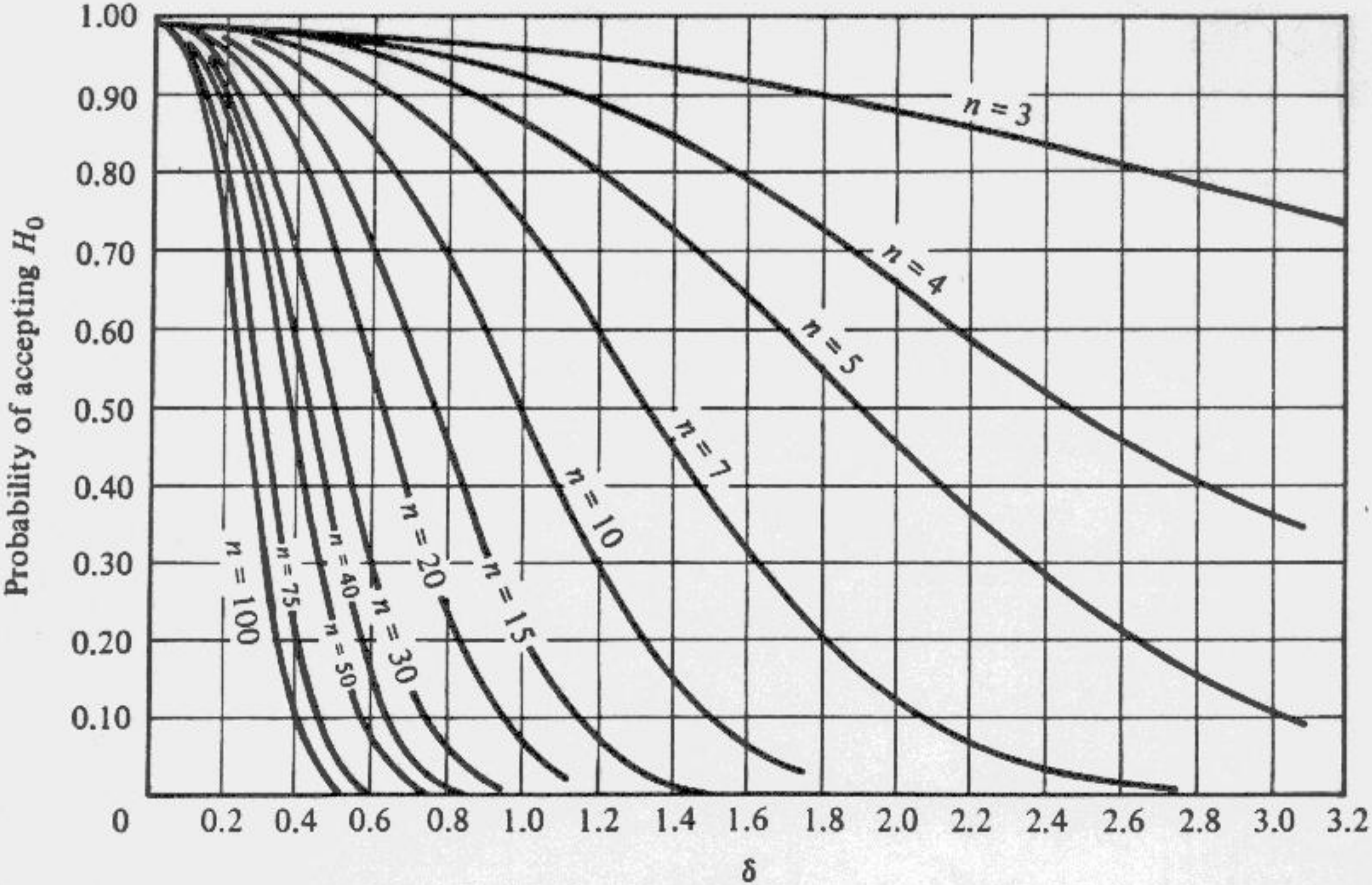
- Use in addition to statistical test, or when no statistical test is readily applicable.

> **Turing Test**
> Described by Alan Turing in 1950. A human jugde 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.


1.0
0.8
0.6
0.4
0.2
0
1
2
3
Probability of accepting H_0
δ
$n = 2$
3
4
5
7
10
15
20
30
40
50
75
100

(a) $\alpha = 0.05$


Probability of accepting H_0
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
δ
n = 3
n = 4
n = 5
n = 7
n = 10
n = 15
n = 20
n = 30
n = 40
n = 50
n = 75
n = 100

(b) α = 0.01