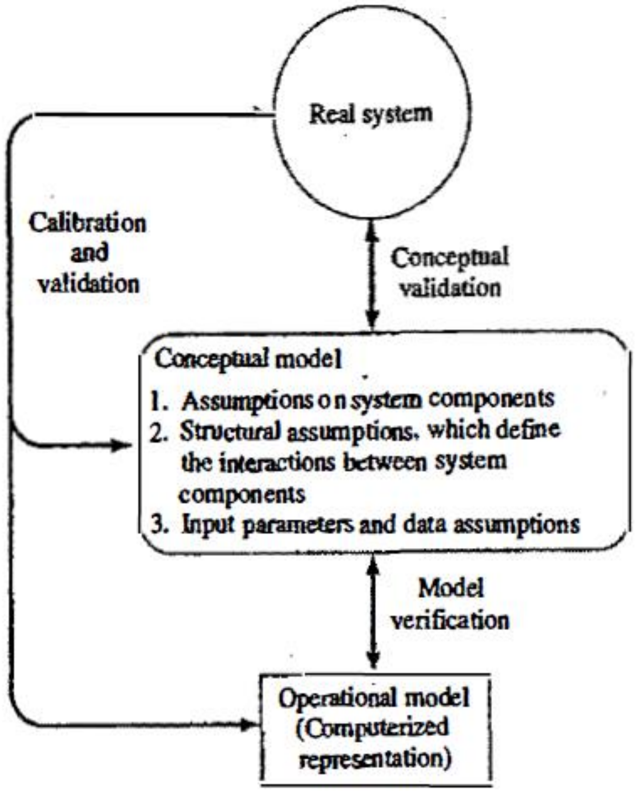
Computer system modeling and simulation

7. Model verification and validation

Sosina M.
Addis Ababa institute of technology (AAiT)
2012 E.C.

Model building

- Observing the real system structure and behavior
  - Collecting data
- Conceptual model
  - Assumptions about the components and structure of tl models, hypotheses about the model input parameters
- Implementation of an operational model
  - Usually by using simulation software



Verification of simulation models

- Goal:
  - Check the correct implementation of the model
- Method
  - Comparison of the conceptual or mathematical model to computer representation
- Check
  - Is the conceptual model implemented correctly in the simulation software?
  - Are the input parameters and logical structure of the model represented correctly?

Verification process

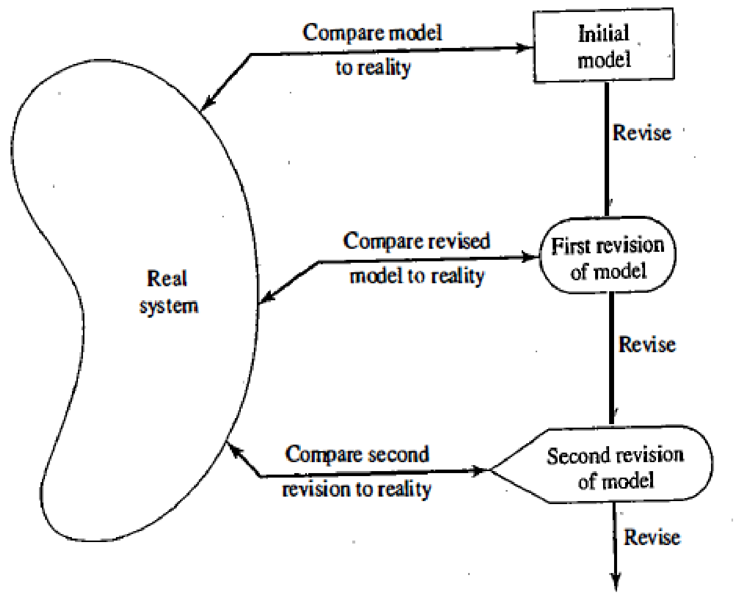
- Make a flow diagram that includes each logically possible action a system can take when an event occurs, and follow the model logic for each action for each event type
- Closely examine the model output under a variety of settings of the input parameters
  - E.g., does an average number in a queue increase with increasing average arrival rate ?
- long-run measures of performance
  - Compare measure of performance that can be computed analytically and its simulated counterpart

Validation of models

- Goal:
  - check the accuracy of the model
  - Increase the credibility of the model
- Method
  - Comparison of the model with the real behavior
  - Model calibration
- Check
  - Is the model an accurate representation of the real system ?

Model Calibration

- An iterative process
  - Compare the model to the actual system behavior
  - Using the discrepancies between the two, improve the model (parameters)
  - Repeat the process until the model accuracy reaches to an acceptable level



Model Calibration

- Use different data sets for validation and calibration
  - after the model has been calibrated by using the original system data set, a "final" validation is conducted, using another system data set.
  - If unacceptable discrepancies between the model and the real system are discovered in the "final" validation effort
    - return to the calibration phase and modify the model until it becomes acceptable.

Validation process

- ***A three-step approach***
  - Build a model that has high face validity
  - Validate model assumptions
  - Compare the model input-output transformations to corresponding input-output transformations for the real system

Validation process

- ***Face validity***
  - Construct a model that appears reasonable on its face to model users and who are knowledgeable about the real system being simulated
- ***Validation of model assumptions***
  - Structural assumptions
    - involve questions of how the system operates and usually involve simplifications and abstractions of reality
    - E.g., consider a customer queueing and service facility in a bank
      - ✓ Customers can form one line or there can be an individual line for each tellers
      - ✓ If there are many lines, customers could be served strictly on FIFO basis or customers could change lines

Validation process

❑ ***Validation of model assumptions***

- Data assumptions
  - Should be based on the collection of reliable data and correct statistical analysis of the data
  - E.g., in the bank study
    - ✓ Interarrival times of customers during several 2-hour periods of peak loading
    - ✓ Interarrival times during a slack period
    - ✓ Service times for commercial accounts and personal accounts
  - Input data analysis
    - ✓ Identify an appropriate probability distribution
    - ✓ Estimate the parameters of the hypothesized distribution
    - ✓ Validate the assumed statistical model by a goodness of fit test

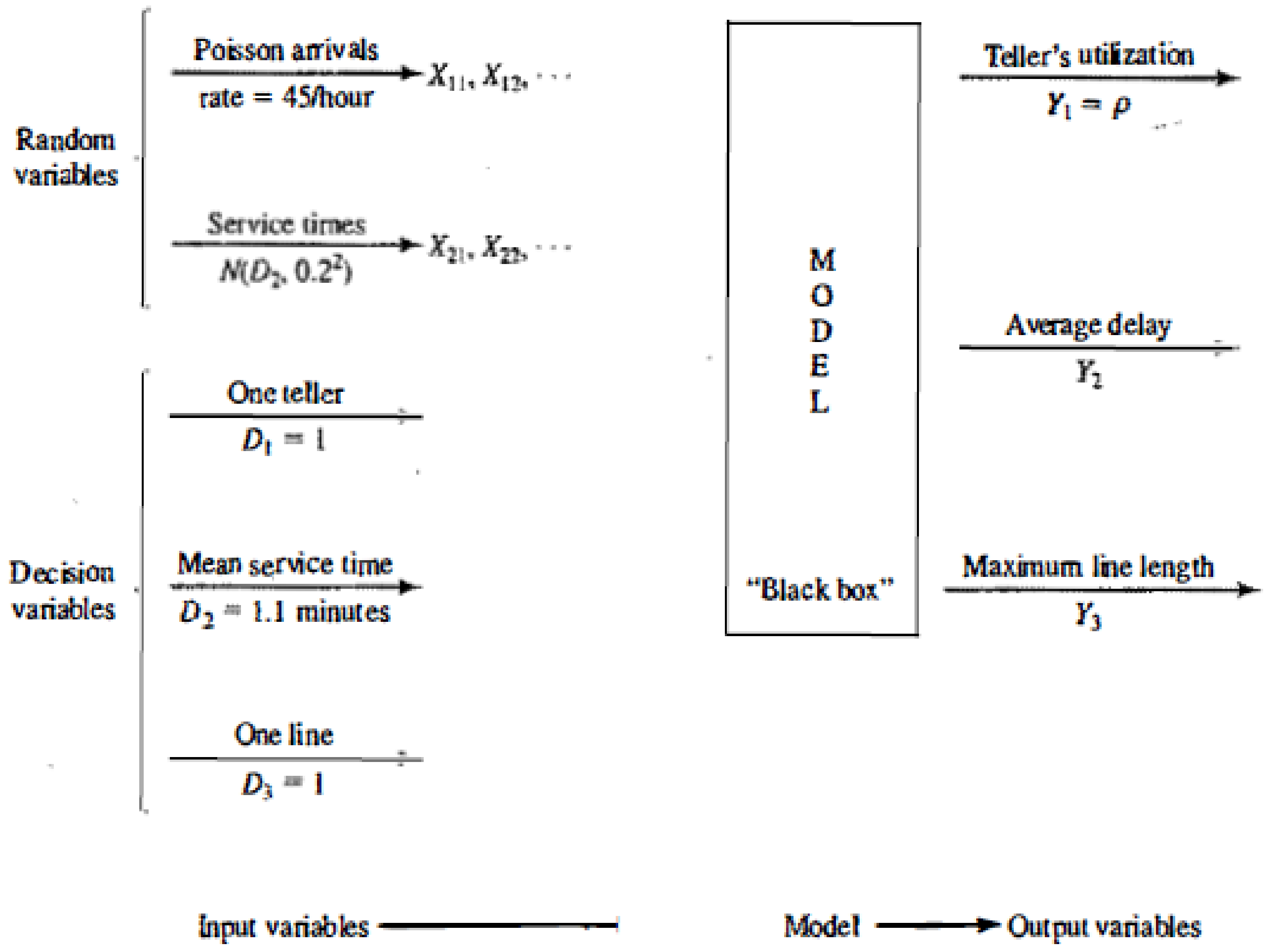
Validation process

- ***Validating input-output transformations***
  - The ultimate test of a model is the model's ability to predict the future behavior of the real system
  - In this phase of the validation process
    - The model is viewed as an input-output transformation, i.e., the model accepts values of the input parameters and transforms these inputs into output measures of performance
  - A historical data is used to test the model
  - Compare the response of the real system and the model under similar input conditions

Validating input-output transformations

- Example: a bank system with one drive-in window serviced by one teller.
  - Service times {S;, i = 1, 2, . . ., 90} and interarrival times {A;; i = I, 2, . .., 90}were collected for the 90 customers who arrived between 1 1 :00 A.M. and 1 :00 P.M. (assumed to be representative of a typical rush hour)
  - Arrivals are modeled as a Poisson process at a rate of 45 customer/hour and the service times were approximated normally distributed with mean 1.1 minutes and SD 0.2 minutes
  - Input variables
    - Interarravial time – exponential with mean 1/45 hr
    - Services times – $N(1.1, (0.2)^2)$

Model input output transformation



Model input output transformation

- For validation of the input-output transformations
  - The system responses should be collected during the same time period in which the input data were collected

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
$X_{11}, X_{12}, \dots$	Y_3 = maximum line length
Service times, $N(D_2, 0.2^2)$	Other output variables of
$X_{21}, X_{22}, \dots$	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 times
$D_3 = 1$ (one line)	
	Y_7 = average length of waiting line

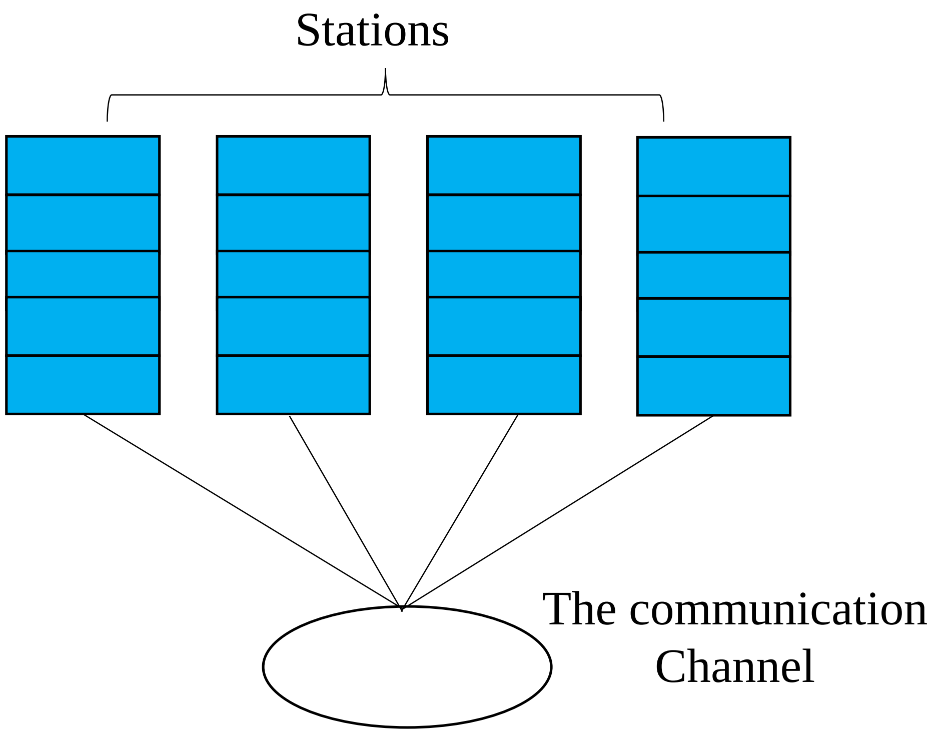
Simulation of computer system

Model of a CSMA system

- Consider a CSMA (Carrier Sense Multiple Access) system
- Stations communicate over a shared channel
- A simple CSMA
  - Sense the channel before transmitting
  - If the channel is idle
    - Transmit the packet
  - If the channel is busy
    - Defer the transmission for a random amount of time

Model of a CSMA system

- Entities
  - Transmitting stations
  - The channel
- What are we interested in?
  - System throughput
  - Collision probability
  - Average queue length
  - Access delay, etc.



Model of a CSMA system

- State variables
  - The state of the channel
  - The number of packets queued in each station
- Events
  - Packet arrival
  - End of transmission
  - Collision