Simulation Examples and General Principles

Basic Steps

- **Determine the characteristics of the inputs.**
  - Usually modeled as probability distributions
- **Construct a simulation table.**
  - Problem specific
  - Often consists of a set of inputs and one of more responses
  - Repetitions
- **Repeatedly generate a value for each input and evaluate the function.**

Simulation Examples and General Principles

- **Simulation examples (chap. 2)**
  - Concrete & specific cases in various areas
  - Example:

    1 + 2 + 3 + …. + 100 = ?
- **General principles (chap. 3)**
  - Generalized representations and concepts
  - Example:

    i, n, S

    $1 + 2 + 3 + \ldots + n = \sum_{i=1}^{n} \ldots$

Simulation of Queuing Systems

A queuing system consists of:

- **Calling population.**
  - *Often infinite: a unit leaves, there is no change in the arrival rate.*
- **Arrivals.**
  - *Random.*
- **Service mechanism.**
  - *A unit will be served in random length based on a probability distribution.*
- **System capacity.**
  - *No limit.*
- **Queuing discipline.**
  - *Order of service, e.g., FIFO.*

Queuing Systems

- **Arrivals and services are defined by the probability distribution of time between arrivals and the distribution of service times, respectively.**
- **Service rate vs. arrival rate**
  - Unstable or explosive
- **State**
  - Number of units in a system and status of server
- **Event**
  - Stimulus that causes system state to change.
- **Simulation clock**
  - Trace of simulation time.

Single-Channeling Queuing System

- **What are the entities?**
- **What are the states?**
- **What are the events? When do events occur or how to model events?**
- **How do the states change when the events occur?**

Other Examples

- **The Able Baker Carhop Problem: Two channels.**
- **Inventory Systems**
  - Simulation of an *(M, N)* inventory system.
- **Reliability Problem**
  - Evaluation of alternatives
- **Military Problem**
  - Random Normal Numbers
- **Lead-Time Demand**
  - Histogram

Summary of Simulation Table Based Approach

- **Basic simulation concepts:**
  - Determine the characteristics of the input data.
  - Construct a simulation table.
  - Generate random variables from the input models & calculate values of the response.
  - Analyze the results.
- **Main problem with simulation table approach:**
  - Can't deal with complex dependencies.

General Principles in Discrete-Event Simulation

Goal: *Dynamic* & *stochastic* systems change in discrete manner.
Definitions:

- **System**
- **Model**
- **System state**
- **Entity**
- **Attributes**
- **List or queue**
- **Event**
- **Event Notice: A record of an event, along with data (time and event type).**
- **Event List or Future Event List (FEL): a list of event notices for future events.**
- **Activity: A duration of specified time (service time, interarrival time, processing time)**
- **Delay: A duration of unspecified time, not known until it ends. Often, an output.**
- **Clock: Simulated time**

Activity and Delay

- **Activity**
  - Also called unconditional wait.
  - Computable from the specification when it begins.
  - An event notice is created having an event time = the activity duration time, cause completion of an activity is an event (primary event).
- **Delay**
  - Also called conditional wait.
  - Not specified ahead of time.
  - Determined by system conditions.
  - Entity involved is placed on a list or queue.
  - Completion of a delay is called a secondary event, no event notice is created.
  - Often is one of the outputs.

Example (Able and Baker)

- **What are the simulation components?**
  - System state
  - Entities
  - Events
  - Activities
  - Delay
- **Components are static. Need to model system dynamics – relationships & interactions between components.**
  - How does an event affect/trigger other system components?
  - How are activities defined?
  - What are the initial and termination system states?

Simulation Process

- **A sequence of system snapshots representing the evolution of the system through time.**
- **A snapshot at a given time includes:**
  - System state
  - List of activities and status of components when each activity ends.
  - Cumulative statistics.

Event Scheduling

The essential idea: move along the time scale until an event occurs and then, depending on the event, modify the system state and possibly schedule new events.

Based on this, it is a trivial exercise to run through a simulation of the system. The events are stored in an event queue, which lists all events in order.

The first event in the queue is taken off, and other events may then be added (assuming that an event only triggers other events in the future).

Event Scheduling cont'd

- **Arrival Event:**
  1. Check the status of the server(s) (idle or busy)
  - If there is an idle server
    - Update idle status; Start serving the customer.
    - *Generate a new departure event at t + s** for this customer.
  - If all busy, place customer in queue & update LQ(t)

  2. *Generate new arrival event t + a** **for next customer.**
  3. Collect stats; Return control to time-advance routine

- **Departure Event**
  1. Check queue (empty or not)
  - If empty, set server to idle.
  - If not empty then do
    - Choose a waiting customer, start serving the customer
    - *Generate a new departure event at time t + s** for this customer

  2. Collect stats; Return control to time-advance routine

Event Scheduling / Time Advance Algorithm

- **The sequence of actions which a simulator must perform to advance the clock and build a new system snapshot.**
- **Future event list (FEL): a key element**
  - Contains all event notices for events that have been scheduled to occur at a future time.
  - Advances simulation time and guarantees that all events occur in correct order.
  - What does scheduling a future event mean?
  - Figures 3.1 and 3.2.
- **Examples**

Activity Scanning

- **In this model, there are three activities:**
  1. An arrival occurs
  2. A service is completed
  3. A service begins
- **The actions associated with these activities are**
  1. (Arrival) Put customer in queue, generate next arrival
  2. (Completion) Declare server idle
  3. (Begin) Make server busy, remove customer from queue, generate completion.
- **There is one new activity: the beginning of service. It is not triggered by any other activity. Instead, it occurs when the following two conditions are satisfied:**
  1. There are customers in the queue
  2. A server is idle

Activity Scanning (cont'd)

- **We call such an activity a conditional activity (also called a C activity or conditional event). As such, the system must be continually scanned for the satisfaction of the conditions of the activity.**
- **What is the advantage of this approach? Mainly, it is simpler to think about and formalize. When first asked to generate the events of a simulation, many people come up with events that require activity scanning.**
- **The main disadvantage with this approach is inefficiency. It is difficult to get the computer to always scan for such activities without slowing down the simulation. To date, the disadvantages of such an approach out way the advantages.**

Process-Interaction

- **A process describes the life cycle of an entity, including events, activities, and delays as the entity moves through a system.**

  Example: a typical process includes time-sequenced list of arrival, delay, begin service, activity, end service.
- **Simulation software provides tools to define the system.**

  Example: a SOURCE to generate arriving customers, a QUEUE to hold waiting customers, and a FACILITY to serve customer.
- **From a modeler point of view, it's simpler. Because events being scheduled on a future event list are hidden.**