Pramod Parajuli
Simulation and Modeling, CS-331

Chapter 9
Discrete System Simulation

Introduction

Event

Change in the state of the system

Discrete Systems

- Systems having discrete nature
- Can be of two types
  1. Fixed time step – a timer or clock exists
  2. Event-to-event (discrete event simulation) – the event works as the clock i.e. self clocking

     (figure 1.2, Law & Kelton, page 9)

The state of the system is not altered while the transition is being occurred.

Clock time

Number recorded time

Introduction

Attributes

Properties of a given entity

Event notice

A record of an event to occur at the current or some future time, along with any associated data necessary to execute the event

Event list

A list of event notices for future events, ordered by time of occurrence; also known as the future event list (FEL)

Delay (conditional wait)

A duration of time of unspecified indefinite length which is not known until it ends.

Introduction

Activity (unconditional wait)

A duration of time of specified length which is known when it begins

Completion of an activity is an event (primary event)

Completion of a delay is conditional/secondary event. Such events are not represented by event notices nor do they appear on the FEL

Duration of activity

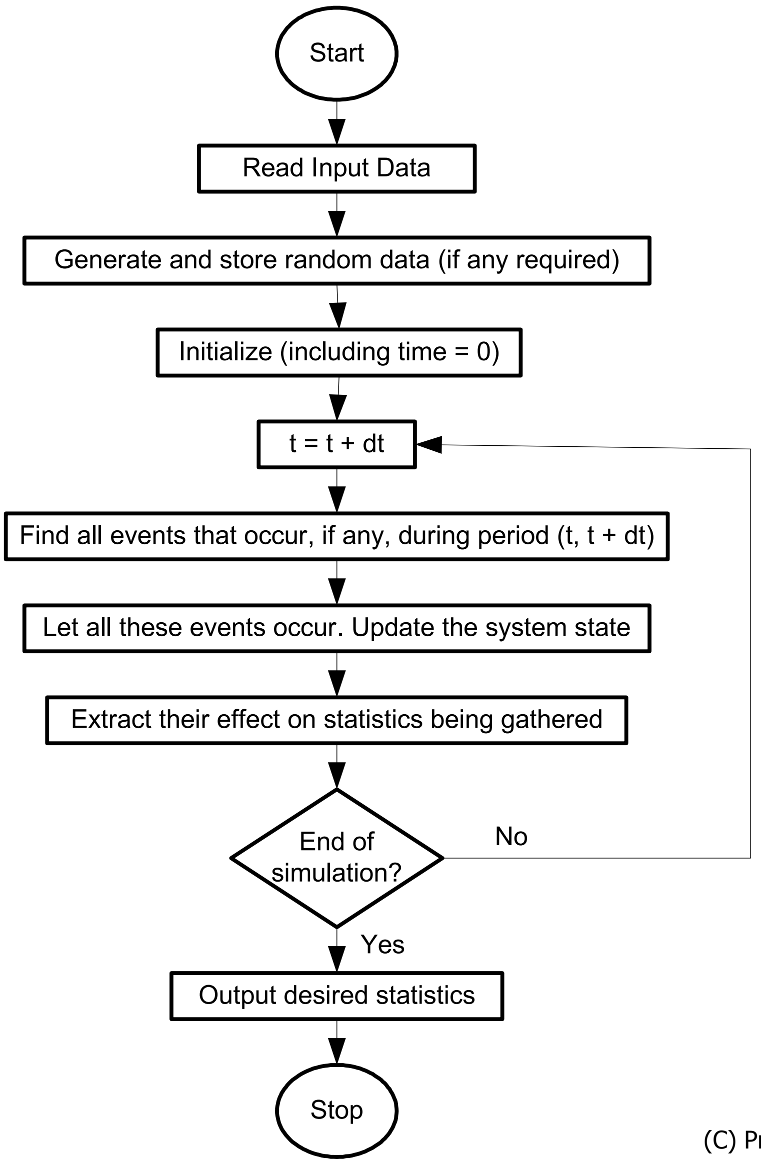
- Deterministic (given exact duration)
- Statistical (random draw from specified set with equal probability of picking an element)
- Functional (defined by a function of any kind)

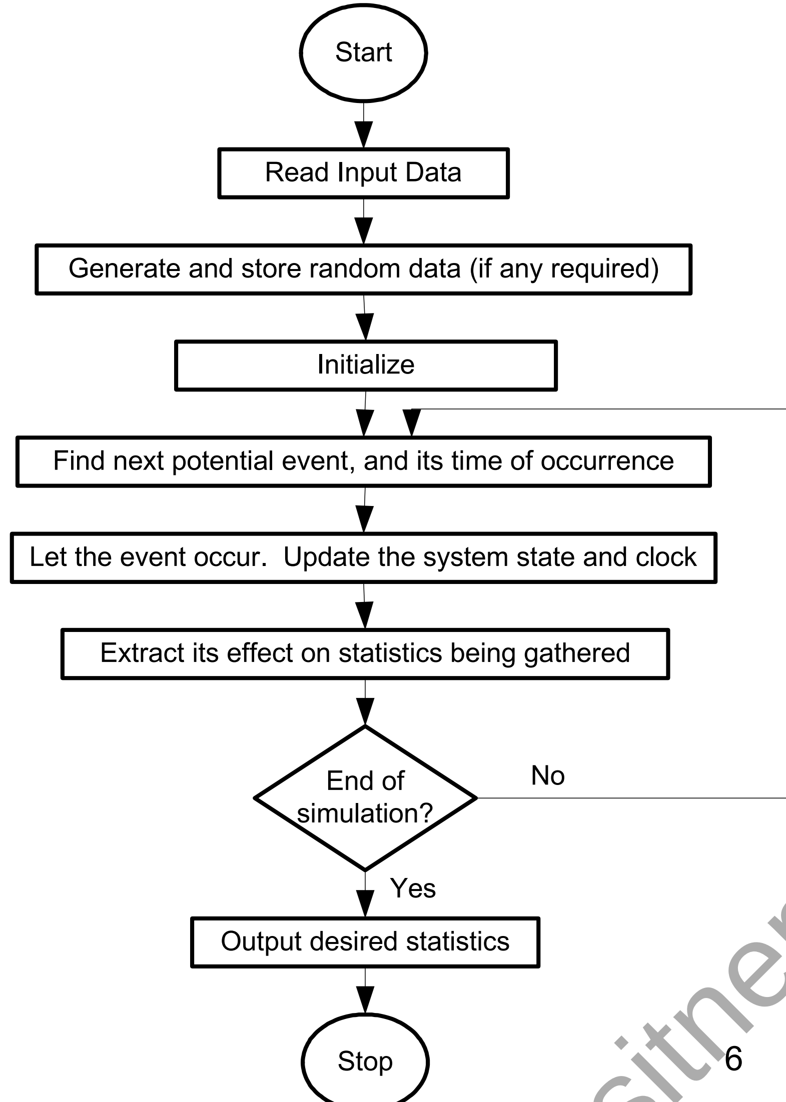
Introduction

Significant event simulation

- Continuous systems with quiescent periods (interval between events)
- Low order polynomials are used to represent the quiescent periods

Fixed time-step vs. Next-event





Components of next-event time-advance approach

- System state
  - The collection of state variables necessary to describe the system at a particular time
- Simulation clock
  - A variable giving the current value of simulated time
- Event list
  - A list containing the next time when each type of event will occur
- Statistical counters
  - Variables used for storing statistical information about system performance
- Initialization routine
  - A subprogram to init the simulation model at time 0

Components of next-event time-advance approach

Timing routine

A subprogram that determines the next event from the event list and then advances the simulation clock to the time when that event is to occur

Event routine

A subprogram that updates the system state when a particular type of event occurs i.e. one event routine for each event type

Library routines

A set of subprograms used to generate random observations from probability distributions that were determined as part of the simulation model

Components of next-event time-advance approach

Report generator

A subprogram that computes estimates (from the statistical counters) of the desired measures of performance and produces a report when the simulation ends

Main program

A subprogram that invokes the timing routine to determine the next event and then transfers control to the corresponding event routine to update the system state appropriately.

Generation of Arrival Patterns

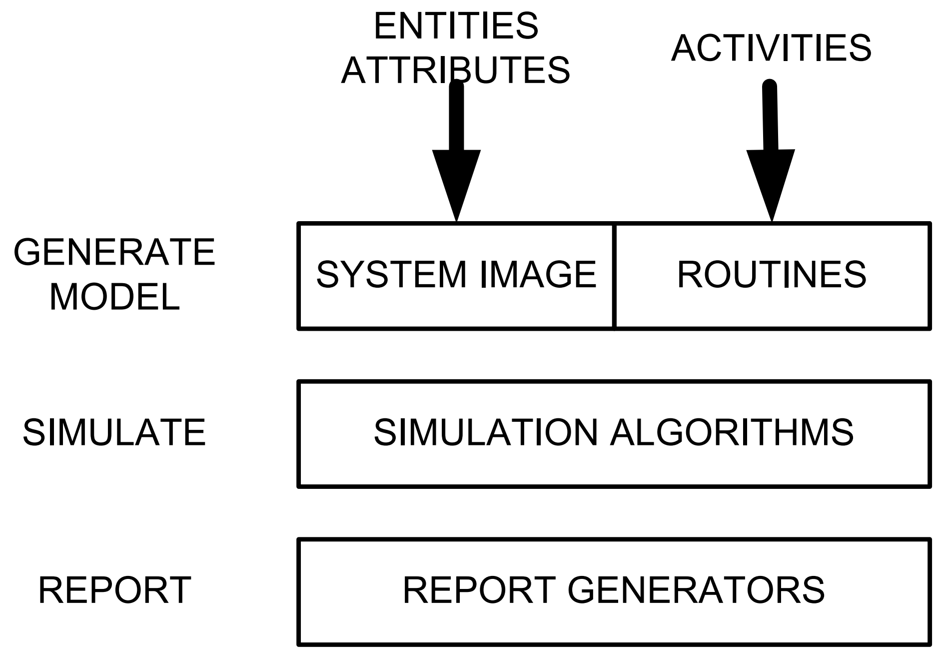
- Generation of exogenous arrivals
- Trace driven simulation - the sequence of inputs may also be generated from observations on a system
- If there is no interaction between the exogenous arrivals and the endogenous events of the system, then the creation of a sequence of arrivals in preparation for the simulation is permissible otherwise arrivals are generated as required
- Arrival of exogenous entity is an event and the arrival time of the next entity is recorded as one of the event times
- The event of entering the entity into the system is executed and the arrival time of next entity is calculated from inter-arrival distribution

Generation of Arrival Patterns

- ***Bootstrapping*** – the previous process of counting the arrival of next entity as current entity is executed is known as bootstrapping. It is a process of making one entity create its successor.
- In bootstrapping, the simulation program have to keep record of arrival of next entity only. Therefore, it is the most preferred method of generating arrivals.
- The attributes for arriving entity can be assigned at the generation time or later on depending on the requirement.
- If the attributes do not affect the event execution, then the attributes can be assigned later on.
- But if attributes do affect the event execution, they must be assigned at the arrival time and retained until the event execution is finished.

Simulation Programming Tasks

- 3 basic tasks
    1. Generate model
    2. Simulate
    3. Report



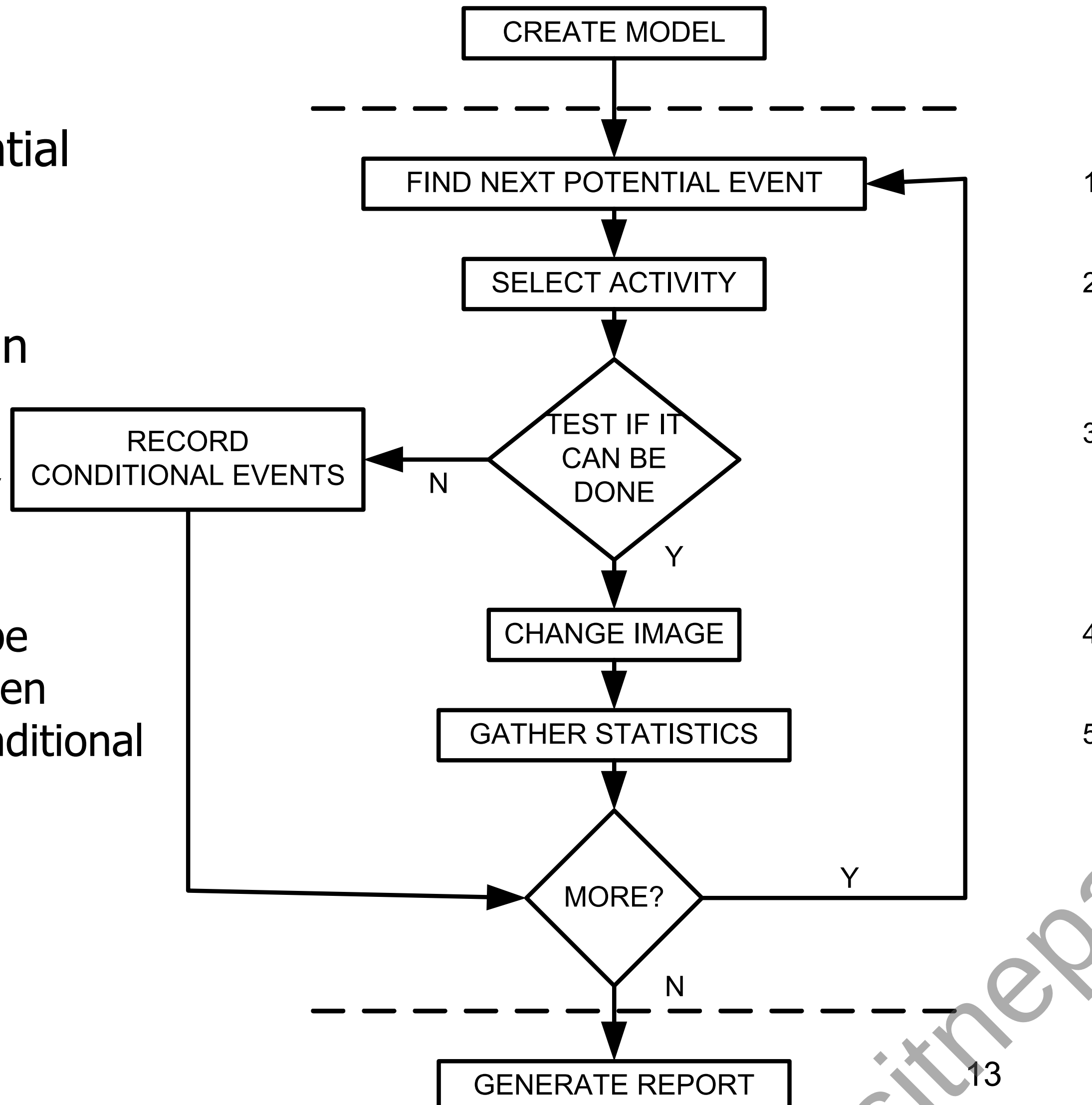
System image – Set of numbers created to represent the state of the system

Simulation algorithms – procedures that execute the cycle of actions

Simulation Algorithm

Steps

1. *Find* the next potential event
2. *Select* an activity
3. *Test* if the event can be executed
   - If direct measure, either success or abandon
   - If the event can be executed later, then it is known as conditional event
4. *Change* the system image
5. *Gather* statistics



Gathering Statistics

Commonly used statistics parameters (included in report);

1. *Counts* – no. of entities of particular type or no. of times some event occurred
2. *Summary measures* – extreme values, mean values, standard deviations
3. *Utilization* – fraction of time some entity is engaged
4. *Occupancy* – fraction of a group of entities in use on the average
5. *Distributions* – queue lengths, waiting times etc.
6. *Transit times* – time taken for an entity to move from one part of the system to some other part

Counters and Summary Statistics

- Counters used to accumulate totals, some level etc.
- Mean of N observations x_i,

$$mean = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- Standard deviation

$$s = \left[\frac{1}{(N-1)}\sum_{i=1}^{N}(m - x_i)^2\right]^{1/2}$$

Measuring Utilization and Occupancy

Measuring the load on some entity is most common requirement of a simulation

What fraction of the time the item was engaged during the simulation run?

Measured by *utilization* factor

Let time t_b be the time the item becomes busy, and t_f be the time when the item becomes free

(fig. 8-11) Gordon

The interval t_f - t_b is added to a counter. At the end, the utilization is derived as;

$$U = \frac{1}{T} . \sum_{r=1}^{N} (t_f - t_b)_r$$

Measuring Utilization and Occupancy

A discrete simulation program, updating time as events occur, will measure the intervals $t_f - t_b$ directly

A continuous simulation program updating time in small intervals need to build up the count by counting the number of intervals in which the item is busy. For example, if for 'x' number of counts item was busy then the fact must be preserved.

If the items was busy at the end of the simulation, then add a count according to the amount of time left in the current count as the trail of previous run

Same holds for if the item was busy at the beginning of the simulation run

For a group of such items; $$A = \frac{1}{T} . \sum_{r=1}^{N} n_r . (t_{r+1} - t_r)$$

Measuring Utilization and Occupancy

If there is an upper limit on the number of entities, then the occupancy indicates the average number in use as a ration to the maximum

If 'M' is the maximum number of entities, and the quantity nr is the number busy in the interval t_r to t_{r+1}, the average occupancy, assuming the number n, changes N times is;

$$= \frac{1}{NM} \sum_{r=1}^{N} n_r (t_{r+1} - t_r)$$

Utilization – timing information for each individual entity

Occupancy – count of a class of entities and last time the count changed

For large number of active entities, utilization cost more (time and space) as compared with occupancy

Recording Distributions and Transit Times

To determine the distribution of a variable, we need to count the no. of times the value of the variable falls within specific intervals

For each new value, it is compared with predefined intervals and for that interval, counted up once

All information is kept in a table that requires;

1. The lower limit of tabulation (L)
2. The interval size (dx)
3. The number of intervals (N)

(Fig. 8-13 Gordon)

To measure transit times, the clock is used in the manner of a time stamp i.e. each entity holds time stamp. When entity reaches a point from which transit time is started, the arrival time is noted. Later when the entity reaches to end point, the difference is computed