Discrete Event Simulation

Discrete event simulation concerns the modeling of a system as it evolves over time by a representation in which stat variables change instantaneously at separate points in time. Here event is defined as an instantaneous occurrence that may change the state of the system. Discrete event simulation can be done by hand calculation. But to solve real world systems data must be stored to simulate it on a digital computer.

Eg:- A service facility with a single server. A one operator barbershop or an information desk at an airport. Here we would like to estimate the average delay of arriving customers where the delay in queue of a customer is the length of time interval from the instant of his arrival at the facility to the instant he begins served. To calculate the average delay of a customer, the state variables in a discrete- event simulation depends upon the condition of server i.e. either the server is busy or idle, the number of customers waiting in queue to be served and the arrival time of customer. Arrival time of customer is needed to compute is delay. This time is given by the service time minus arrival time. Here in this system two types of event occurs:

i) The arrival of customer
ii) Completion of service of customer

Arrival is an even as it changes the status of server from idle to busy and increases the number of customer in queue.

Similarly departure is an event since it changes the state of server from busy to idle and decreases the number of customer in queue

Components and organization of a Discrete-Event Simulation Model

1) System State- The collection of state variables necessary to describe the system at a particular time.
2) Simulation Clock- A variable giving the current value of simulated time.
3) Event list- A list containing the next time when each type of event will occur.
4) Statistical Counters- Variables used for storing statistical information about system performance.
5) Initialization Routine- A subprogram that initialize the simulation model at time 0.
6) Timing Routine- A subprogram that determines the next event from the event list and then advances the simulation clock to the time when the event is to occur.
7) Event Routine- A subprogram that updates the system state when a particular type of event occurs.

8) Library Routines- A set of subprograms used to generate random observations from probability distributions that were determined as part of simulation model.
9) Report Generator- A sub program that computer estimates of the desired measures of performance and produces a report when the simulation ends.
10) 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. It also checks the termination and invokes the report generator when the simulation is over.

Generation of Arrival Patterns

Arrival pattern for particular system is specified for simulation. The exogenous arrivals can be designed for simulation.

Trace Driven Simulation

The sequence of inputs can be generated from the observation on a particular system are tested from record gathered from a running system, i.e. representative of the sequence of operations the computer system will have to execute. This method is trace driven simulation. Here program monitors can be attached to the running system to extract data with no or little disturbance to running system.

Bootstrapping Method

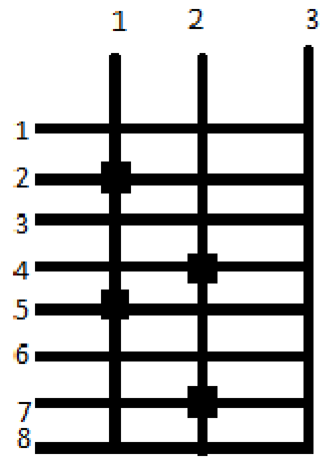
The arrival time of an entity is recorded as one of the event times. When simulation clock time reaches this event time, the event entering in the system is executed and the arrival time of next entity is calculated. This method is called bootstrapping method. Here one entity create its successor.

Telephone call Simulation as lost call System

This system represents telephone system where call gets lost when call cannot be connected at the time of its arrival. Call gets lost in two condition. When line is not available i.e. destination call is busy call gets lost and the other condition is when link is not available. Link is the connection between two lines.

Let us make some assumption to simulate the system

1. Let n= no of total link then line=2*n+2
2. Arrival process is bootstrapping method (simulation clock is updated at every arrival that may be either call connect or call disconnect)



Let us show the various state of simulation for following telephone calls. Here recently 2 is calling 5 and 4 and 7 is calling. In the various system state lines shows the status of lines. Lines those are busy are indicated by 1 and the line those are free are indicated by 0. Links indicates the maximum number of link and link in use. Call in progress indicates the status of ongoing call. Processed, completed, block and busy gives the counter of calls. Next arrival indicates the arrival of next call.

Lines
1-0
2-1
3-0
4-1
5-1
6-0
7-1
8-0

Clock	12

Next arrival		
From	To	Length
1	7	20
Arrival time-20		

Call in progress		
From	To	End
4	7	35
2	5	18

Links	
Max	3
In use	2

Processed	Completed	Block	Busy
2	0	0	0

System state-1

In system state 2 the, when the simulation clock reaches time 18, call 2-5 is completed and thus they are removed from call in progress, their state is change to 0 from 1, link in use is decremented by 1 and counter of call completed is increased by 1.

Lines
1-0
2-0
3-0
4-1
5-0
6-0
7-1
8-0

Links	
Max	3
In use	1

Clock	18

Call in Progress		
From	To	End
4	7	35

Next arrival		
From	To	Length
1	7	20
Arrival time-20		

Processed	Completed	Block	Busy
2	1	0	0

System state-2

In the system state 3, when simulation clock reached 20, line 1 tries to call line 7 which is still in call in progress and the connection is abandoned. In this case call counter of busy is incremented by 1 and all other remains same. Here call gets lost since line is busy.

Lines
1-0
2-0
3-0
4-1
5-0
6-0
7-1
8-0

inks	
Max	3
In use	1

Clock	20

Call in progress		
From	To	End
4	7	35

Next arrival		
From	To	Length
3	6	15
Arrival time-25		

Processed	Completed	Block	Busy
3	1	0	1

System state-3

Different states goes in the similar manner. If the line is busy call gets lost else till the link is available call connects. When call in progress is full that is max link = link in use, then call gets block as in the case of system state 6.

Lines
1-0
2-0
3-1
4-1
5-0
6-1
7-1
8-0

Links	
Max	3
In use	2

Clock	25

Call in progress		
From	To	End
3	6	40
4	7	35

Next Arrival		
From	To	Length
2	8	15
Arrival time-28		

Processed	Completed	Block	Busy
4	1	0	1

System state-4

Lines
1-0
2-1
3-1
4-1
5-0
6-1
7-1
8-1

Links	
Max	3
In use	3

Clock	28

Call in progress		
From	To	End
2	8	43
3	6	40
4	7	35

Next Arrival		
From	To	Length
1	5	15
Arrival time-30		

Processed	Completed	Block	Busy
5	1	0	1

System state-5

In this state since all the links are full, call cannot be connected though the lines are free. In this case also call gets lost and the call counter of block is increased by 1.

Lines
1-0
2-1
3-1
4-1
5-0
6-1
7-1
8-1

Links	
Max	3
In use	3

Clock	30

Call in progress		
From	To	End
2	8	43
3	6	40
4	7	35

Next Arrival		
From	To	Length
1	5	15
Arrival time-45		

Processed	Completed	Block	Busy
6	1	1	1

System state-6

Delayed call system

Let us modify the telephone call system so that the call that cannot be connected at the time of arrival does not get lost. They wait until they cannot be connected. The system is like a message passing switching system with store and forward capability.

To record of delay it is necessary to build another list like call in progress. For this we create another list, delay call list as shown in the system state 1. The first two states are same as lost call system.

Lines
1-0
2-1
3-0
4-1
5-5
6-0
7-1
8-0

Links	
Max	3
In use	2

Clock	12

Next arrival		
From	To	Length
1	7	15
Arrival time-20		

Call in progress		
From	To	End
4	7	25
2	5	15

Delay Call list		
From	To	Length

Processed	Completed	Block	Busy
2	0	0	0

System state-1

Lines
1-0
2-0
3-0
4-1
5-0
6-0
7-1
8-0

Links	
Max	3
In use	1

Clock	15

Next arrival		
From	To	Length
1	7	15
Arrival time-20		

Call in progress		
From	To	End
4	7	25

Delay Call list		
From	To	Length

Processed	Completed	Block	Busy
2	1	0	0

System state-2

Now, when call is completed it is necessary to check the list call for waiting call. If there is any call the call is forwarded to call in progress. If there is not any next arrival call is processed. At simulation clock 20, 1 calls 7 which is busy till clock time 25. Thus it is stored in delay call list at shown in system state 3.

Lines
1-0
2-0
3-0
4-1
5-0
6-0
7-1
8-0

Links	
Max	3
In use	1

Clock	20

Next arrival		
From	To	Length
8	6	20
Arrival time-30		

Call in progress		
From	To	End
4	7	25

Delay Call list		
From	To	Length
1	7	15

Processed	Completed	Block	Busy
3	1	0	1

System state-3

The system goes to next state with record of delayed call. At time 25 call from 4 to 7 is completed. Now, 7 is free. Thus call from delayed list is connected which is from 1 to 7. Now, the record are updated. Delayed call is transferred to call in progress list and the system state change like this.

Lines
1-1
2-0
3-0
4-0
5-0
6-0
7-1
8-0

Links	
Max	3
In use	1

Clock	25

Next Arrival		
From 8	To 6	Length 20
Arrival time-30		

Call in Progress		
1	7	40

Delay Call list		

Processed	Completed	Block	Busy
3	2	0	1

System state-4

Similarly block condition can be shown and call connection can be done.

Simulation programming task

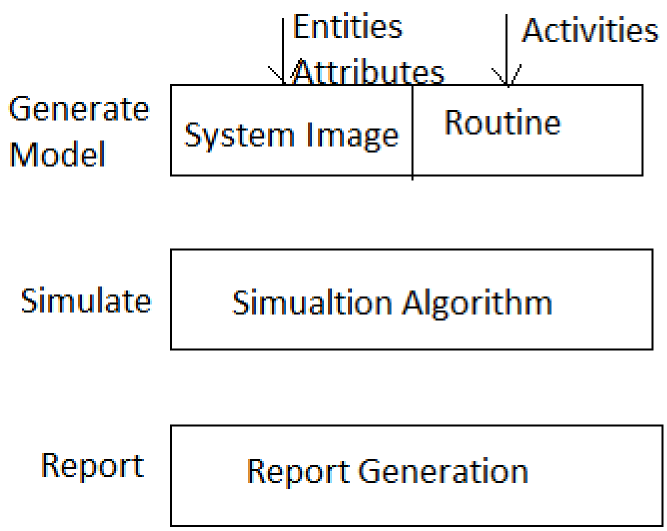
There are three main task to be performed.

i) Generate Model

ii) Simulate

iii) Report

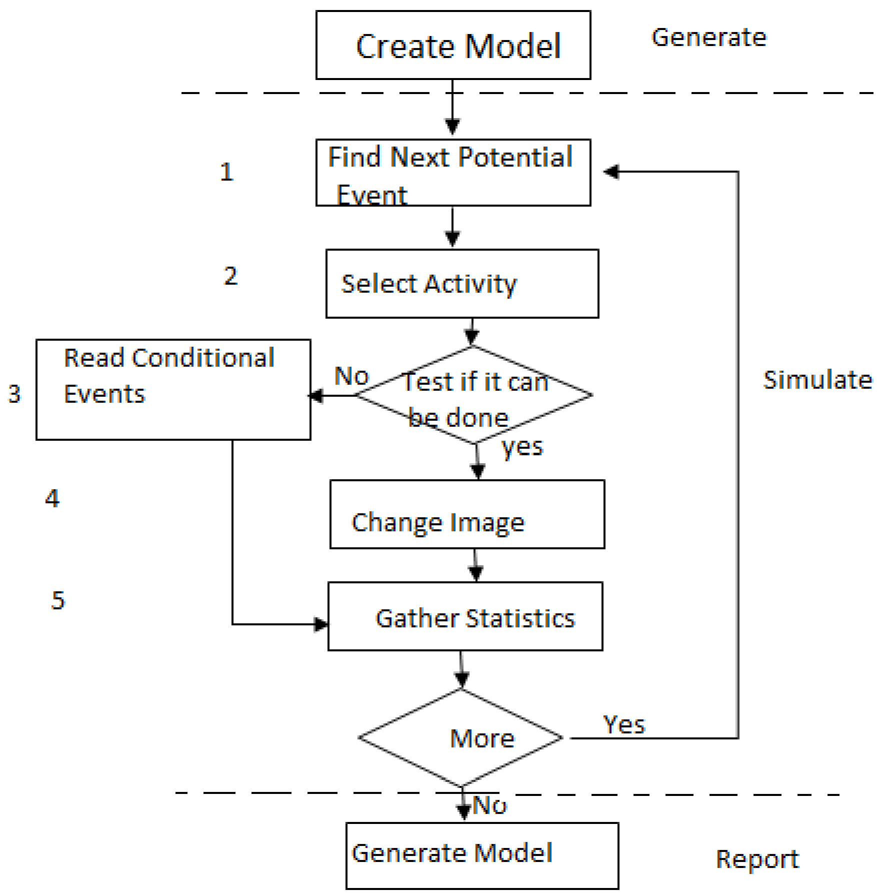
The first is to generate a model and initialize it. From the description we generate system image which is the set of numbers that elects the state of the system all times. The second task is to program the procedure that executes cycle of actions to carry out simulation. This process is simulation algorithm. The third task is report generation.



Simulation Programming Task

To carry out simulation algorithm steps are:

1. Find the next potential event
2. Select an activity
3. Test if the event can be executed
4. Change the system image
5. Gather statistics



Flow chart

Gathering Statistics

Commonly required statistics are:-

1. Counts – gives the number of entities or number of times event occurred.
2. Summary Measures- gives extreme values, mean values and standard deviations.
3. Utilization- time some entity is engaged.
4. Occupancy- groups of entities used on average.
5. Distributions- variables such as queue length or waiting times.
6. Transit times- defines as time taken for an entity to move from one part of the system to other part

Counters and Summary measures-

Counters are the basis for most statistics. Some accumulates total and some record the current values in the system. The telephone system simulation use counters to record the total number of lost and busy calls. Maxima and minima are easily obtained.

The mean of a set of N observations $X_r = 1, 2,3,\ldots\ldots, N$

$$m = 1/N \sum_{r=1}^{n} X_r$$

Standard deviation is defined as

$$s = \sqrt{\frac{1}{N-1}\sum_{r=1}^{N}(m - x_r)^2}$$

Common method is

$$\sum_{r=1}^{N}(m - x_r)^2 = \sum_{r=1}^{N} x_r^{\ 2} - Nm^2$$

Measuring utilization and occupancy

Utilization is used to describe what fraction of time the item is engaged during simulation run. To measure utilization, record of the time t_b is necessary which gives the time at which the item last become busy. When the entity becomes free at time t_f the interval $t_f - t_b$ is added to a counter. At the end of simulation run, the utilization U is derived by dividing the accumulated total time T for entity used N times.

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

For discrete system $t_f - t_b$ is measure directly. For continuous system count is given by counting the number of intervals in which the item is busy. Fro group of entities, it also requires the information about numbers of entities involved.

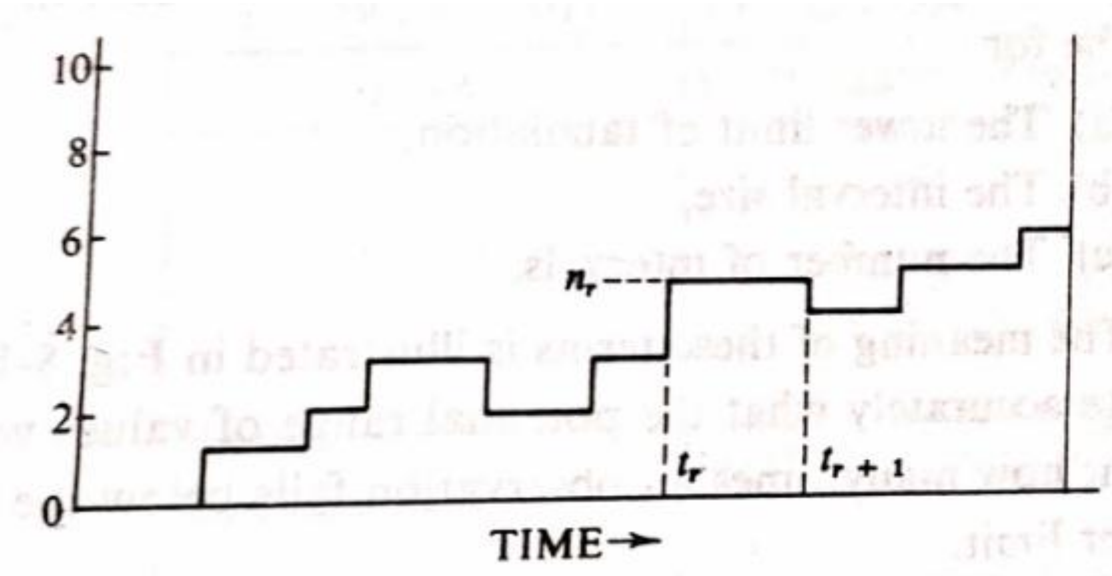


Figure: time history of busy telephone links

Above figure represents as a function of time, the number of links in a telephone system that are busy. To find average no of links in use record of currently used links and time of last change should be kept. If number of changes at time t_r to the value n_r then at the time of next change t_{r+1} ,the quantity $n_r (t_{r+1}-t_r)$should be calculated and accumulated to total. Therefore for run A at time T

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

If there is an upper limit on the number of entities, as there was a limit on the links in telephone system, the term occupancy is used to describe the average number in use as a ratio to the maximum. Thus, if there are M links in a telephone exchange and quantity n_r is the number busy in interval t_r to t_{r+1} , the average occupancy assuming the number n_r changes N times is

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

For utilization timing information is needed and for occupancy a count of class of entities and record of count that changed last time is required.

Recording Distribution and Transit Times

To record the distribution of variable, a count is required that gives the value that the variable falls within specific intervals. A table is set which gives the location to record the values for specific interval and accumulate each count. For new observation the value is compared with the limits established for intervals and 1 is added to counter.

The tabulation intervals are uniform in size and contains

a) The lower limit
b) The interval size
c) The number of intervals

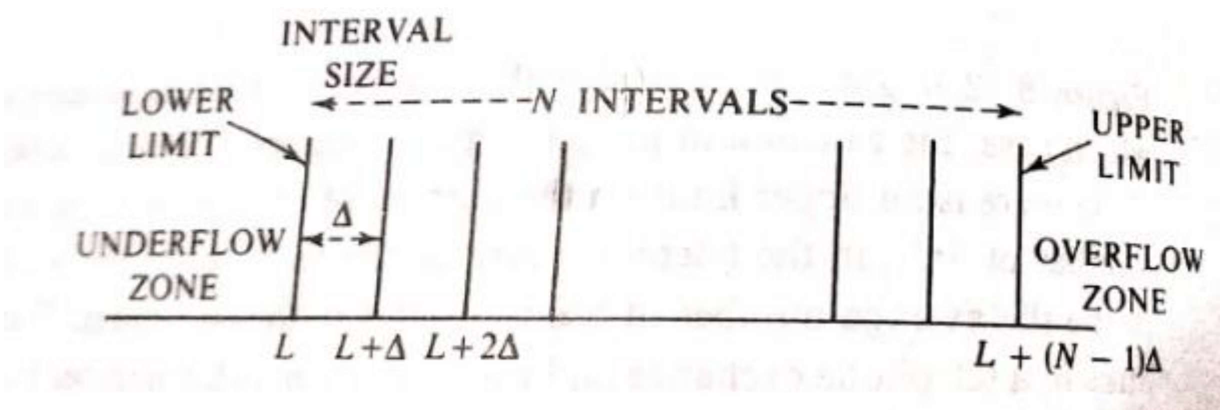


Figure: Definition of a distribution Table

It is necessary to count how many times an observation falls below the lower limit and beyond the upper limit. To calculate mean value and the standard deviation, number of observation is accumulated and the squares of the observations is added at the same time of distribution. Rach observation x_i increase a count to 1 in the counter and total x_i is Σx_i and the squared sum is Σx_i^2.

To measure transit time, the clock is used in the manner of a time stamp. When and entity reaches a point from which a measurement of transit time is to start, a note of arrival time is made. Later, when the entity reaches the point at which the measurement ends a note of clock time upon arrival is made and compared with the first time to derive the elapsed interval.