Chapter 4

#discrete system simulation:

Discrete event simulation concerns the modeling of a system as it

#REPRESENTATION OF TIME:

The passage of time is recorded by a number referred as clock time. Its usually set to zero at the beginning of simulation and sub-sequently indicated how many units of simulated time have passed since the beginning of simulation. Unless specifically stated otherwise, the term simulation, time means the indicated clock time and not the time that computer has taken to carry out the simulation. As a rule, there is no direct connection between simulated time and the time to carry out the computation.

To basic methods exist for updating clock time. One method is to advance the clock to the time at which the next event occur. The next method is to advance the clock by small(usually uniform) intervals of time and determine at each interval whether and event is due to occur at that time. The first method is referred to as event oriented and the second method is said to be interval oriented.

Discrete system simulation is usually carried out by using the event oriented method while continuous system simulation normally uses the interval oriented method.

#GENERATION OF ARRIVALS PATTERNS:

Arrival pattern for particular system is specified for simulation . the exogenous arrival can be designed for simulation. The sequence of input can be generated from the observation on particular system. The components of system are tested from records gathered from a running system .i.e. representative of sequence of operations the computer system while have to execute. This simulation is trace driven simulation.

Here program monitors can be attached to the running system to extract data with no or little disturbance to the running system. The arrival time of an entity is recorded as one of the even times. When simulation clock times reaches this event time, the event of system is executed and the arrival time of next entity is calculated. This method is called bootstrapping. Here one entity creates its successor.

#SIMULATION OF TELEPHONE SYSTEM:

The simulation of a discrete system can be explained by simulating a telephone system.

The telephone system has several telephone(here only 8 are shown) connected to a switchboard by line. The switchboard has several lines provided the condition that only one connection at a time can be made to each time. Any call that cannot be connected at the time it arrives is immidetly abandoned and then the system is called a lost call system. A call may be lost due to following reasons:

1. If the called party is engaged(busy)
2. If the no link is available(a blocked call).

links

Lines **1 2 3**

|  |  |  |
| --- | --- | --- |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |

Horizontal= telephone link=n=3

Vertical= links lines=2^n =2^3=8

**(\*)**Object of simulation(lost call system):

To process a given number of calls and determine what portion is successfully completed, blocked or found to be busy calls. Lets consider the current state as below:

Lines

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

|  |  |
| --- | --- |
| Max no | 3 |
| In use | 2 |

|  |
| --- |
| Clock |
| 1027 |

|  |  |  |
| --- | --- | --- |
| From | To | Length |
| 1 | 7 | 20 |

Next call

|  |
| --- |
| Arrival time |
| 1057 |

Call in progress

|  |  |  |
| --- | --- | --- |
| From | To | End |
| 4 | 7 | 1075 |
| 2 | 5 | 1053 |

Call counters

|  |  |  |  |
| --- | --- | --- | --- |
| Project | Completed | Blocked | Busy |
| 131 | 98 | 5 | 28 |

**fig: system state 1**

* In the above table, 0 mean the line is free and 1 means line is free. To keep gtrace of event cock time is included. Currently the clock time is 1027. Here unit of time is one second. The clock updates at the next most event as simulation proceeds
* Each call is entity. Its attributes are origin, destination, length and time at which call finishes.
* There is a list of call in progress showing which line is connected and call finish time.
* To generate arrival of calls bootstrapping is used
* Call can come from any line .i.e not busy at the time of arrival.
* The origin, destination and call length is generated at the time of arrival. The time of next arrival is 1057.
* The records are set for clock time 1027
* There are two activities causing the events, new call can arrive and existing call can finish.
* In fig, there are three future events, the lines between 2 and 5 with finished time 1053, the call between 4 and 7 with finish time 1075 and a new call arrival at 1057. This call will call from line one to seven for 20 sec
* Simulation proceed by executing cycle of steps to simulate each events. In first step, thenext potential event is scanned. in this case the event is at .1053. thus clock is updated to clock time 1053. In second step the activity is selected that cause the event. In third the event execution Is tested. In fourth the records are changed to reflect the effect of events.

Lines

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

|  |  |
| --- | --- |
| Max no | 3 |
| In use | 1 |

|  |
| --- |
| Clock |
| 1053 |

|  |  |  |
| --- | --- | --- |
| From | To | Length |
| 1 | 7 | 20 |

Next call

|  |
| --- |
| Arrival time |
| 1057 |

Call in progress

|  |  |  |
| --- | --- | --- |
| From | To | End |
|  |  |  |
| 4 | 7 | 1075 |

Call counters

|  |  |  |  |
| --- | --- | --- | --- |
| Project | Completed | Blocked | Busy |
| 132 | 99 | 5 | 28 |

***fig: system state 2***

* *call is disconnected by setting 0 to line 2 and 5. After that necessary static are gathered for simulation output and then simulation is continued.*

Lines

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

|  |  |
| --- | --- |
| Max no | 3 |
| In use | 1 |

|  |
| --- |
| Clock |
| 1057 |

|  |  |  |
| --- | --- | --- |
| From | To | Length |
| 3 | 6 | 98 |

Next call

|  |
| --- |
| Arrival time |
| 1063 |

Call in progress

|  |  |  |
| --- | --- | --- |
| From | To | End |
|  |  |  |
| 4 | 7 | 1075 |

Call counters

|  |  |  |  |
| --- | --- | --- | --- |
| Project | Completed | Blocked | Busy |
| 133 | 99 | 5 | 29 |

***fig: system state 3***

***when 1 calls 7, its busy because 7 is in busy state***

* *for clock 1057 , the next event arrives, the clock is updated to arrival time and attributes of new arrival are generated. Here test is carried out to check the availability of link and line since seven is busy the call is lost.*

***(\*)DELAYED CALL SYSTEM:***

*Now let us assume the telephone system is modified that the calls that cannot be connected are not lost. They wait until they cannot be connected. This system is like the message passing switching system which store and forward capabilities. To keep the record of delay calls, it is necessary to built another list like call in progress. The first two steps are same as before lost call system. The system state at clock 1057 when call from line one arrive, the system sate changes as.*

Lines

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

|  |  |
| --- | --- |
| Max no | 3 |
| In use | 2 |

|  |
| --- |
| Clock |
| 1027 |

|  |  |  |
| --- | --- | --- |
| From | To | Length |
| 1 | 7 | 20 |

Next call

|  |
| --- |
| Arrival time |
| 1057 |

Call in progress

|  |  |  |
| --- | --- | --- |
| From | To | End |
| 4 | 7 | 1075 |
| 2 | 5 | 1053 |

Call counters

|  |  |  |  |
| --- | --- | --- | --- |
| Project | Completed | Blocked | Busy |
| 131 | 98 | 5 | 28 |

**fig: system state 1**

Lines

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

|  |  |
| --- | --- |
| Max no | 3 |
| In use | 1 |

|  |
| --- |
| Clock |
| 1053 |

|  |  |  |
| --- | --- | --- |
| From | To | Length |
| 1 | 7 | 20 |

Next call

|  |
| --- |
| Arrival time |
| 1057 |

Call in progress

|  |  |  |
| --- | --- | --- |
| From | To | End |
|  |  |  |
| 4 | 7 | 1075 |

Call counters

|  |  |  |  |
| --- | --- | --- | --- |
| Project | Completed | Blocked | Busy |
| 132 | 99 | 5 | 28 |

***fig: system state 2***

|  |  |  |
| --- | --- | --- |
| from | to | End |
|  |  |  |
| 1 | 7 | 20 |

Lines delayed calls

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

|  |  |
| --- | --- |
| Max no | 3 |
| In use | 1 |

|  |
| --- |
| Clock |
| 1057 |

|  |  |  |
| --- | --- | --- |
| From | To | Length |
| 3 | 6 | 98 |

Next call

|  |
| --- |
| Arrival time |
| 1063 |

Call in progress

|  |  |  |
| --- | --- | --- |
| From | To | End |
|  |  |  |
| 4 | 7 | 1075 |

Call counters

|  |  |  |  |
| --- | --- | --- | --- |
| Project | Completed | Blocked | Busy |
| 133 | 99 | 5 | 29 |

**Fig: system state 3 (system state 1 and 2 are same as lost call)**

**#GATHERING STATISTICS:**

Commonly required statistics in discrete simulation system are

1. **count:**

it gives the number of entities or no of times event occur.

1. **Summery measures:**

It gives extreme values like mean values and standard deviation.

1. **Utilization:**

It is defined as the fraction or percentage of time of some entity is engaged.

1. **Occupancy**

Its defined as the fraction of group entities used on average.

1. **Distribution:**

its used for variables such as quee length or waiting time.

1. **Transit time:**

Its defined as time taken for an entity to move from one part of the system to other part