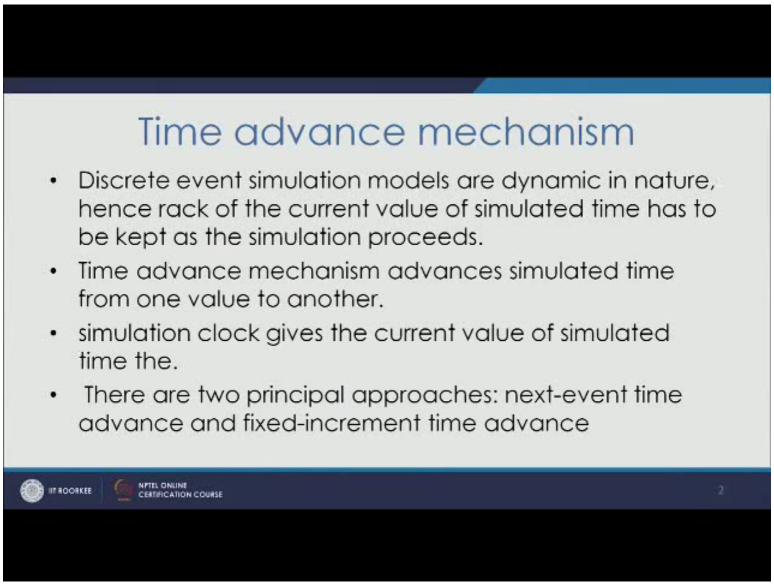
Modeling & Simulation of Discrete Event Systems

Dr. Pradeep K Jha
Department of Mechanical and Industrial Engineering
Indian Institute of Technology, Roorkee

Lecture – 03
Time Advance Mechanism, Components of a Simulation Model

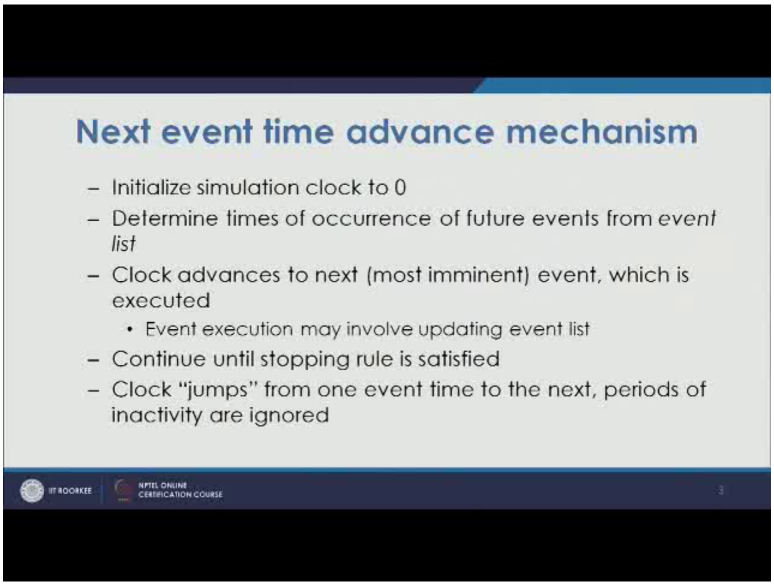
Welcome to the lecture on time advance mechanism and components of a discrete event simulation model. So, so far, we have discussed something about the introduction of discrete event simulation model, and in this lecture, we are going to discuss about the time advance mechanism. As we know in this case when we formulate the problem we make the programs; in the programs because we have already discussed that it is a case of dynamic system. So, the time varies and you are studying over time. So, there are 2 ways by which this time advancement is studied. So, in that as we see discrete event simulation models are since they are dynamic in nature. So, you will have to keep track of the current value which is getting simulated. So, during the process of simulation as the event occurs and as the system state is updated the time changes. And this time are to be updated are to be recorded.

(Refer Slide Time: 01:33)



Now, this time advance mechanism advances the simulated time from one value to the other. So, as the event occurs there will be certainly the it is happening at a different time, and the simulation clock will go at that particular time. Simulation clock gives the current value of simulated time. So, basically this simulation clock will give you the current value of the simulation time. And in fact, there are 2 approaches to see the time advance mechanism. One is next event time advance mechanism, another is fixed increment time advance mechanism. So, we will discuss about these 2 mechanism by which this simulation clocks are basically updated; one is next event time advance mechanism.

(Refer Slide Time: 02:33)



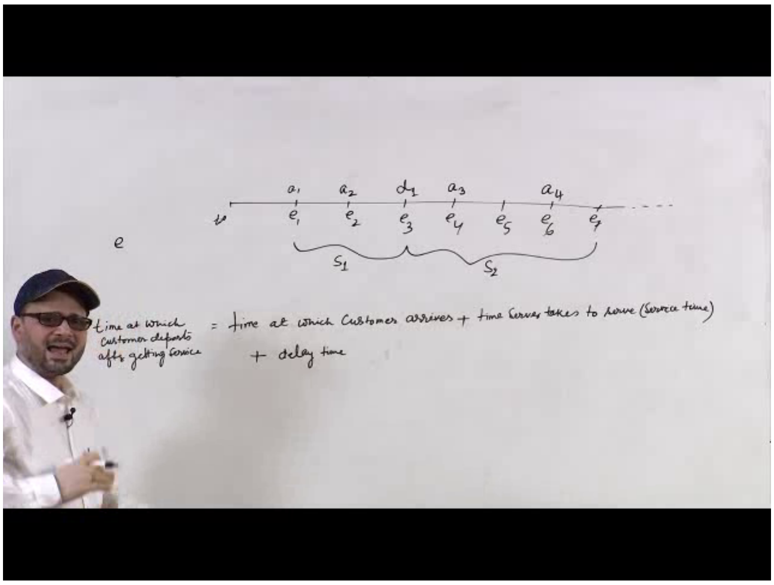
So, in the next event time advance mechanism, we are initialising the simulation clock to 0. As it is written that you initialize this simulation clock to 0. So, when you are starting the simulation, certainly you are normally starting at a 0 time. So, initially it will be at 0 time.

Now, the determine times of occurrence of future events. So, basically you will have the event list the event may be in terms of arrival or departure. So, you will have the event list. From the event list basically, you are picking one of the event. So, once the event is picked, at that time the clock has to move from 0 to that value. So, clock advances to next most eminent event which is executed. So, the thing is that during the process of simulation, this event may be either arrival or departure in case of a queuing model. So, whichever is most eminent initially certainly it is the arrival as we the solution starts, or simulation starts you will have first the arrival. But then in course of time it may be either arrival, or it may be a departure depending upon the service, which goes in the way. So, clock will advance to the next; that is most eminent event.

Event exhibition may involve updating event list. So, certainly as the event list you know gets updated in the simulation process. Because once one event has occurred it will go to the next event, and the next and that basically goes on. Now this continues until some stopping rule is satisfied, for this stopping rule is basically a condition which is the precondition, and that is to be defined when the simulation has to stop, and that maybe depending upon a time frame, or depending upon how many customers are served. So, all that or how many customers are delayed or so. There may be a condition or there will be a condition after wise the simulation has to stop, because it is a program and it will go on doing the it is work. And it will ask whether it should continue or it should stop. So, that stopping condition will be there at one point of time and then simulation will stop.

Clock jumps from one event time to the next and periods of inactivity are ignored. So, in this case what happens, you know the clock will move from one event to the next event. The first event may be occurring at suppose 3 minutes, and second event if it occurs after 10 event, 10 minutes in those case the simulation clock will move from 3 minutes to the 10 minute, because the next event is occurring only at 10 minutes. And this detail has been obtained from the event list. So, this is how this next event time advance mechanism works. Now see; by example we can see how this works. So, suppose you have the time horizon, and in that you are starting from 0 and if event is denoted by e.

(Refer Slide Time: 06:01)



So, suppose you have e 1, e 2, e 3, e 4, e 5, e 6 or e 7, and it will go on. So, these are the event. Now the thing is the simulation clock will start from here the time is 0 here, and it will go the next simulation clock will be at this time. Now this may be either an arrival or a departure. Now the departure will depend upon the service time.

So, suppose this is the first arrival. So, if a is the arrival this will be a 1. So, this is the first arrival. Now this arrival basically the next arrival will depend upon the inter arrival time. So, suppose next arrival may be coming at here. Now this next arrival; when this next arrival will be there that will be basically depending upon this time a 2 minus a 1, or this difference of time. And this value we have to get and as a data and that may be in the form of probability distribution function data or random variates. So now so, what we see is the first arrival is here, and second arrival is here.

Now in this way we may have the third arrival here, we may have the 4th arrival here. The thing is from the data it will take the first arrival at this point; once the simulation clock from t equal to 0, goes to this time it will update all the variable values. So, it will update the measures of performance with in which we are interested in and it will find everything. So, at this point it will see that there is one arrival.

Now, when this one will depart, first departure will depend upon the service time. So, suppose the sever takes this amount of service time, and if it is service time of the first customer. In that case this point will be the departure of the first customer. So, what we see that you have the time at which this first customer has come, then how much it has been delayed. So, the time at which customer arrives, then the time the server takes to serve. So, the time server takes to serve, that is known as service time.

This is also a time which will be basically probabilistic in most of the cases, and we will get from certain distribution function. So, this service time as we see here now then delay; delay time for the first customer suppose we are finding the time at which the customer will depart. That time will be nothing but the time at which the customer comes, the time of service for this customer. And then the time which it experiences delay.

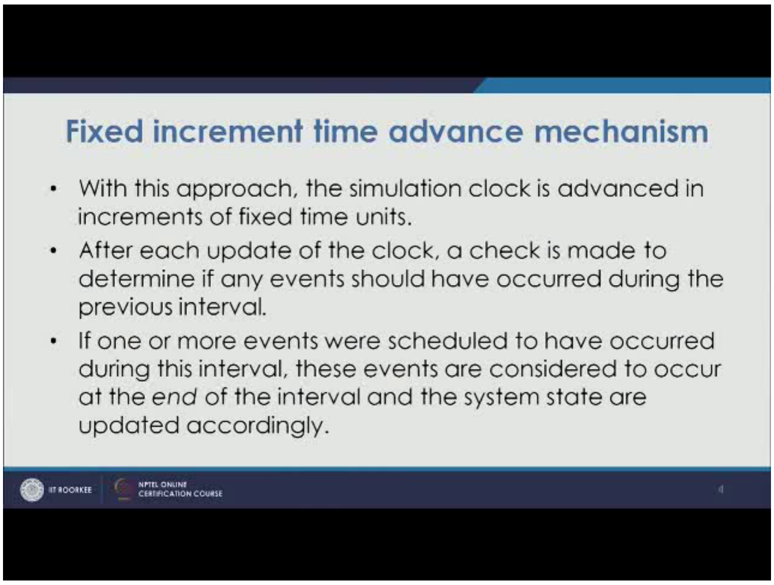
So, in the case of first customer as we see, there is no delay. Because once he has come the server is idle. So, it this will be basically the time at which the time at which customer departs after getting service. So, basically, we can have any suffix I for any customer and that will be nothing but the time at which customer arrives plus a time of service, and then plus the delay time. And that is how we see that this customer will depart at this time that will be d 1.

Similarly, the second customer who which comes here, for them what will happen? If suppose for the second customer the service time is s 2. So, the service will only start when the first customer is going to leave. Now suppose the second service time is this. So, we can see that in this case once the simulation clock reaches here. So, simulation clock will first move from here to here, where one event has occurred. This event is nothing but the arrival of the first customer. So, the clock is advanced to next time e 2 where the second event has occurred and second event is also an arrival. So, that will be a 2.

Then the third event is a departure because this service has been completed and the customer has to depart. So, here it is a third event. And at all these points wherever the event occurs the there will be updation or they will be upgradation of the values; the system state values. So, at this place you will have a departure once there is departure then the next customer who is in queue he will join the queue for getting the services. So, for this customer the second one; second event the arrival of the second customer he will be joining the queue here. So, basically, he joining at this time. So, the time at which customer arrives for second customer will be this time from here to here, then the delay time delay time will be since he is starting in the queue at this time e 3. So, it will be the delay time, and then from here he will start getting the service. So, this will be the service time.

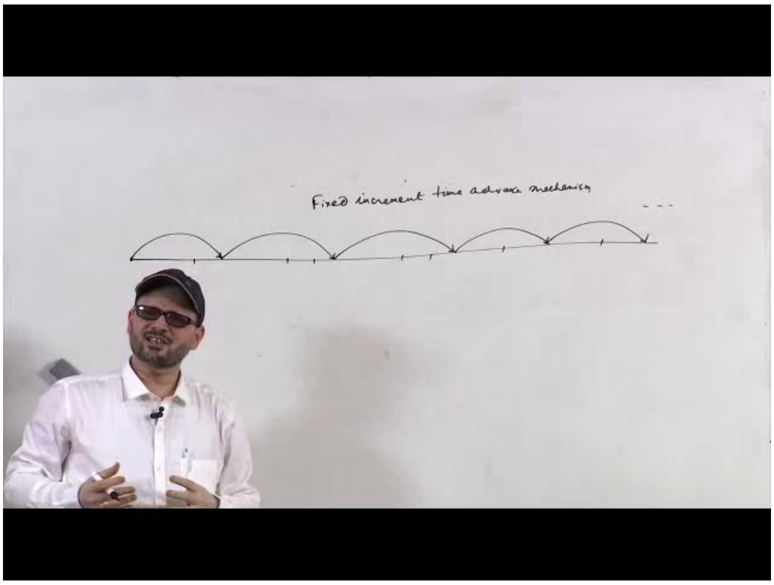
So, in this way here will be the departure of this second customer. In between what we see is there are many points, there may be arrival or departure or so. So, and all those points you will have arrival or departure or so, in that case this fixed event next event time advance mechanism works. Now once we move to another kind of advance mechanism. So, in this case what we see is that whenever there is any event occurring the system state will be updated, you will find all the measures of performance whatever is required to be calculated. Now next is the fixed increment time advance mechanism.

(Refer Slide Time: 14:30)



So, in the case of fixed increment time advance mechanism, the simulation clock will move and it will jump for a fixed increment of time. So, what happened?

(Refer Slide Time: 14:45)



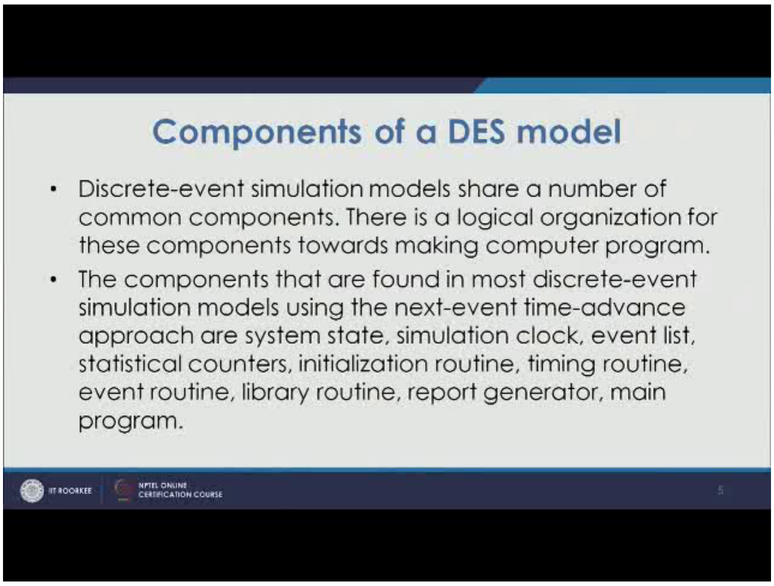
You will have this timeline, and this clock will move in a fixed manner for a constant increment of time.

So, you have this is fixed increment time advance mechanism. Now in this case what happens? The simulation clock will only move to these places, but the event may occur in be occurring at these points. So, what happened? The in this case once the simulation clock will move to this point, it will try to ensure that which of the events have already completed by this time. Because simulation clock is moving for a fixed time increment and then what it does is; at this time, it records or it gives the value of completion of this event. So, basically in this case the simulation clock moves, there are certainly certain disadvantages. And certain inaccuracies in such type of advance mechanism, because we are not representing the event time at it is actual time. We are representing the event time based on the increment time. So, it will be if it is delta t, it will be 2 delta t, it will be 3 delta t, it will be 4 delta t, it will be 5 delta t and so on.

So, in this case for accuracy what you have to do is you can get the accuracy, but for that you may have to have these delta t very small. In those case if the delta t is very small and it captures this event with more accuracy then certainly accuracy will be more, but then having delta t very small value, it will increase the simulation time to a very large value. So, that is why, but then in many practical cases like in economic studies or so. This fixed increment time advance mechanism is adopted, where we analyze the whole years period and then we discuss about the events which have occurred and we feel that at the end of the year every event has occurred.

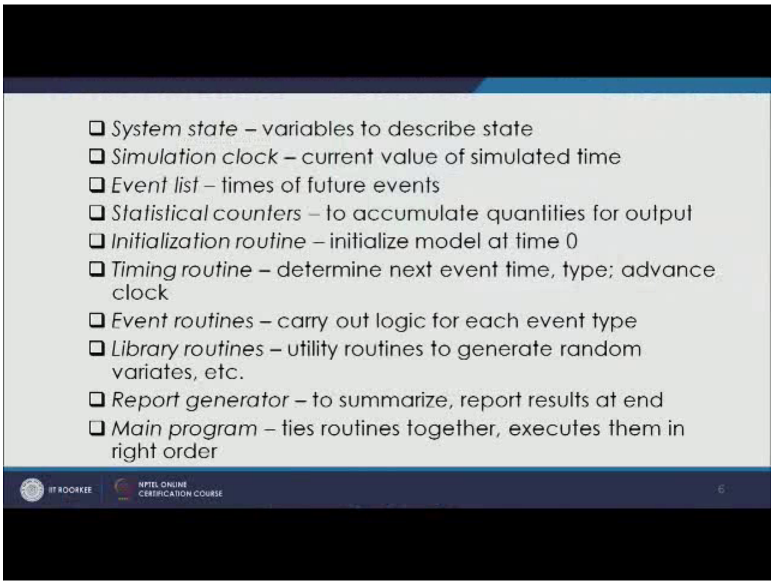
So, that is also followed, but there in in most or the discrete event simulation model softwares the next event type of advance mechanism is followed. Now what are the components of a discrete event simulation model? So now discrete event simulation models share a number of common components and there is a logical organisation for these components towards making a computer program because you have many components.

(Refer Slide Time: 18:09)



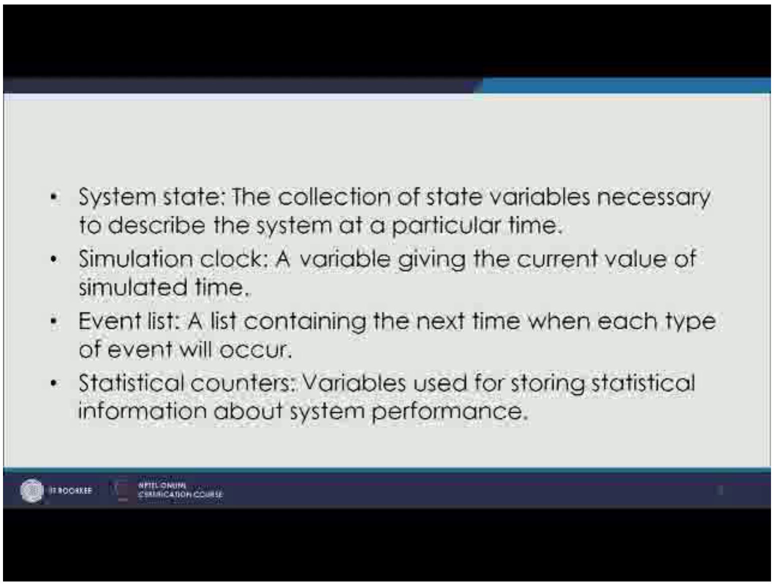
And then they are logically attached, they have some connections. And then they are trying to you are successful in getting a good computer program. We should be able to simulate it. The components that are found in most (Refer Time: 18:26) discrete event simulation models. Using the next event time advance approaches are; one is system state, then you have a simulation clock, then you have event list, a statistical counters, initialisation routine, timing routine, event routine, library routine, report generator and main program. So, we will discuss one by one what is the function of these subroutines. So, the system state.

(Refer Slide Time: 18:58)



The system state it is the variable to describe the state.

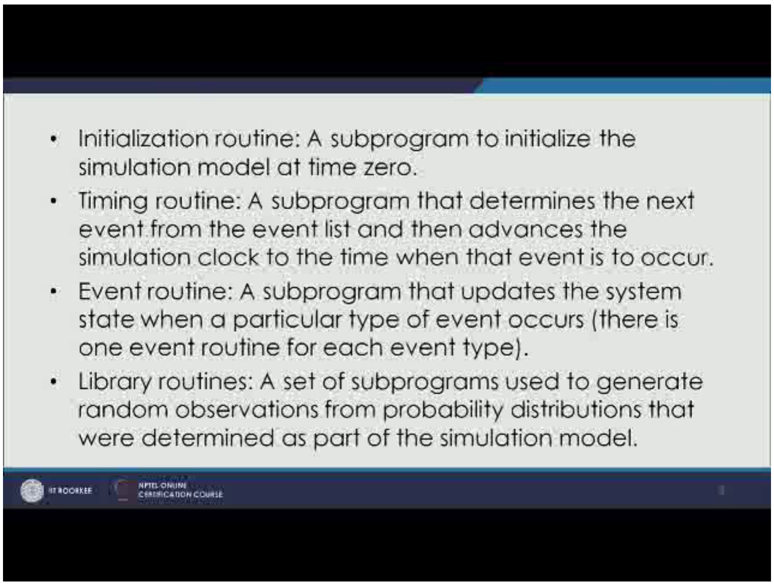
(Refer Slide Time: 19:05)



So, let us see the system state it is; the collection of state variables necessary to describe the system at a particular time. So, as the time proceeds the simulation clock proceeds from one value to next, then the system state with that will be the state variables collection. And it will be trying to describe the system at that particular simulation clock time. Now what is a simulation clock for? So, simulation clock is defined as a variable which is given as the current value of simulated times. So, as we discussed the simulation clock value will move from one to another, and in the case of next event type of mechanism. The simulation clock will move from one to other depending upon what kind of event is coming, what type of event is the next eminent event.

So, simulation clock will move in this fashion. Event list; the event list it is a list containing the next time when each type of event will occur. So, event list will tell you that what is going to come next. As we discussed we have the events like arrival and departure. And normally these arrivals and departures in case of queuing models. It will be either inter arrival time or the time of arrival. So, inter arrival time is normally given which is normally following certain type of probability distribution function, and then you have a service time. So, what will be the service time for the server? So, this will be in the event list. So, that list will be containing. So, this list will be containing the next time when each type of event will be occurring. Statistical counters so, these are the variables used for storing statistical information about system performance. So, as we discussed that our main aim is to analyse the measures of performance, or find the measures of performance. And this is statistical counters they are the variables for storing these statistical information about the systems performance.

(Refer Slide Time: 21:26)



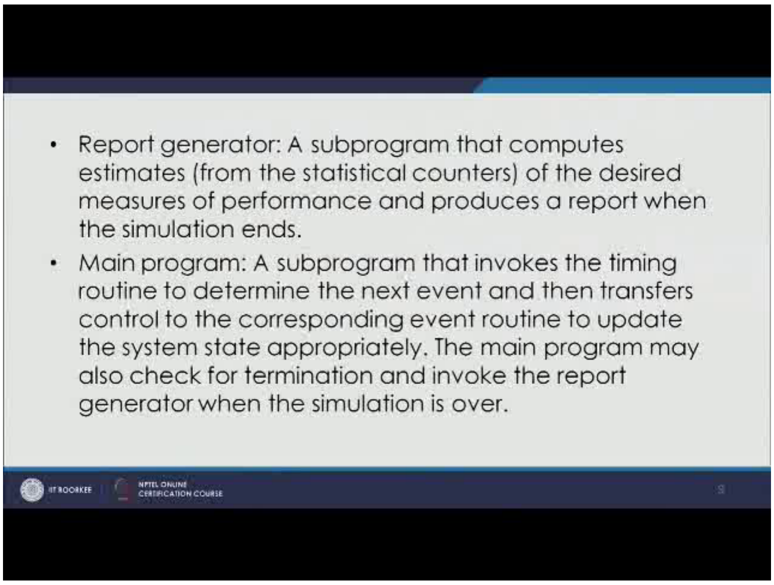
Then the initialisation routine: So, once we start making a program, you will have to initialise the values to 0, and then you have to combine with the main program and then you have to move. So, every time you have the event occurring. So, this at that particular event time it will update all the system state values. And then it will further go and ask the condition whether the simulation is over. So, then if not then it will again go for initialising. So, it because the system clock is advance to the next event time, initialisation routine the sub program to initialize the simulation model at time 0. So, basically since we start at time 0. So, that initialization routine will tell that now the simulation has started. And basically, that goes every time because initialisation of this time will be advanced every time the event occurs. Timing routine: The subprogram that determines the next event from the event list and advances the simulation clock to the time when that event is to occur.

So, it will tell about that next time at which the event is to occur. Similarly, you have event routine the subprogram updates the system state when a particular type of event occurs. So, in this case the it is updating about the event type which occurs. Then you have library routine; this set of subprogram to generate the random variates or random observations. So, basically what happens? In the event list whatever number you are going to get they are basically the probabilistic outcomes, and they are generated from certain set of you know pattern or certain set of distribution functions. So, that basically is seen by or taken care of by these library routines.

So, that basically from probability distribution that were determined as part of the simulation. Because in the simulation model you need to have this data and basically all these data are required and I mean for a particular time and this data is used as the probabilistic value, and it is used for finding the measures of performance. Then you have the report generator. So, basically this report generator that stage will come when the end condition is fulfilled, when you feel that there is end of the program.

So, once the termination condition is satisfied, in that case you are told that is now you make the report. So, this report generator will be the subprogram which computes estimates from the statistical counters of the desired measures of performance and produces a report when the simulation ends. So, it will when the simulation will end it will try to find all the you know statistical counters.

(Refer Slide Time: 24:56)



It will find all the measures of performance values, and then it will tell you that at this event time or at this simulation time this was the situation at as we proceeded further this was the situation and so. So, we will have a report generated; the 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.

So, basically the main program has to invoke or to call 4; 4th the data from this side to the initialisation routine this side. So, the timing routine, then it has to go to connect with the event routine invoke. So, it is main job is to have the invoke; function to invoke the from these subroutines or subprograms and then get the data. So, then it will transfer control to the corresponding event routine to update the system state appropriately. So, it will come to the event routine, it will update everything that event routine will have the you know connection with this library routines. So, from there it will get all these values at which time what event has occurred. And in that case then main program me also check the termination, main program will also see that when there will be termination. So, when the termination condition is satisfied, at that point of time the main program will say that now the program is terminated, and you have to stop working for; stop I mean, simulation. And then it will tell the report generator that you make the report. And once the report is ready then it will go and it will end the simulation.

So, this is how this main program works. So, what we see is; in this case you have a logical flow. Logical flow of you know the information. So, what happens as we will see in the next when we will discuss about the logic how this logical flow occurs, that we have all these subroutines. All these subroutines are to be put and they are they are to have communications with one another. So, initially you will have the initialisation at simulation clock is set to time 0. And then it will have the from the event list it will have the next eminent event type. Then it will ask and it will update all the counter statistical counters system state values. And then it will ask for the termination condition. And then in that case, you are once your condition is satisfied that time; the condition I mean the simulation ends.

So, what we have study this in the earlier part, we had seen that these are the in a nutshell you have a system state which is discussing about the variables to describe a state, simulation clock is giving current value of simulated time. Similarly, all that is given as we see the main program; this main program is time the routines together. And it is helping in executing them in the right order.

It is job will be to see that proper routine is invoked at a proper time. So, that the system state or the variables or whatever we are interested in that is measures of performance calculation they are calculated appropriately. So, so as we see here the library routine will find the random variates. So, it will have the routines to generate these random variates, and basically that helps in predicting these arrivals or the departures. Time routine as we see it will determine the next event time and next event type. And then it will advance the clock to the next time. Initialization routine will be the initialising the model at time 0 and event list.

So, this way you have different kinds of so, different components of this discrete event simulation model. And as we discussed the most of the programs, we have the next event type of advance mechanism being followed. So, as we discuss the different programs like fortran or c where we write or most of the softwares, like you have many computer software which are working for these like arena or so. These all are based on this next event type of time advance mechanism. Otherwise next we cannot say that this next the our fixed increment time advance mechanism is not used. It might be used for different purposes, but then mostly the next event type of advance mechanism is followed. In the national lecture we will discuss about the logic of this program formulation, and also, we will discuss about the steps in a sound simulation study.

Thank you.