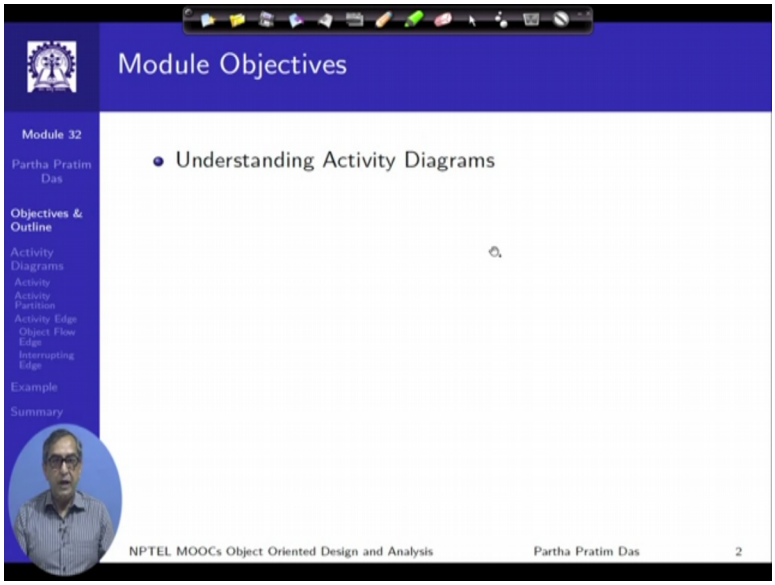
Object-Oriented Analysis and Design
Prof. Partha Pratim Das
Department of Computer Science and Engineering
Indian Institute of Technology-Kharagpur

Lecture – 44
Activity Diagrams Part I

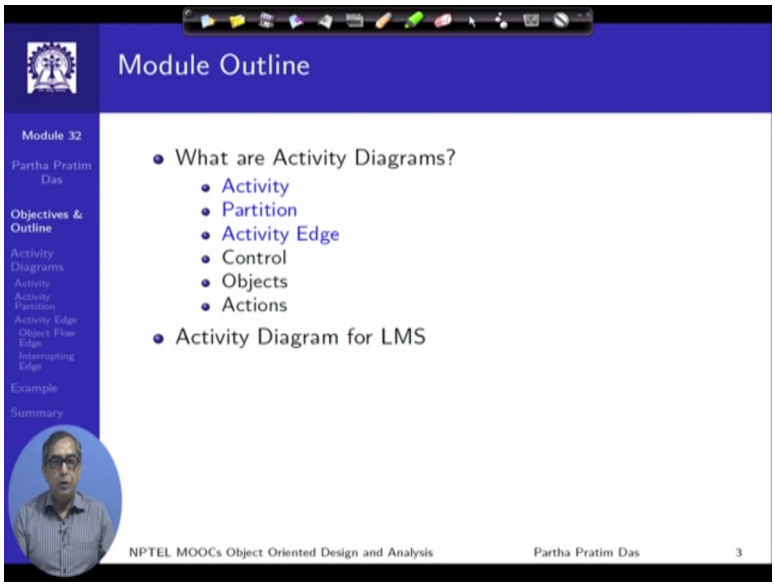
Welcome to module 32 of object-oriented analysis and design. In this module and in the next 2, we will continue we will discuss about the activity diagram of uml.

(Refer Slide Time: 00:34)



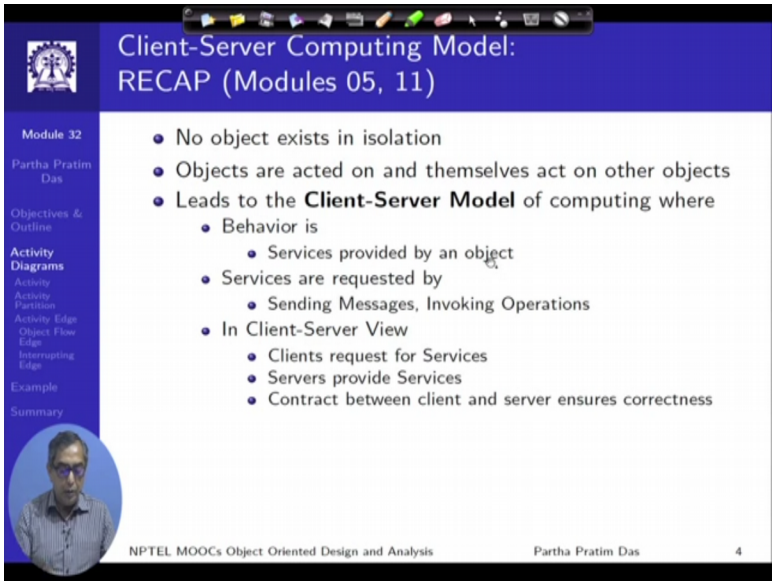
So, the objective would be to understand the activity diagram.

(Refer Slide Time: 00:41)



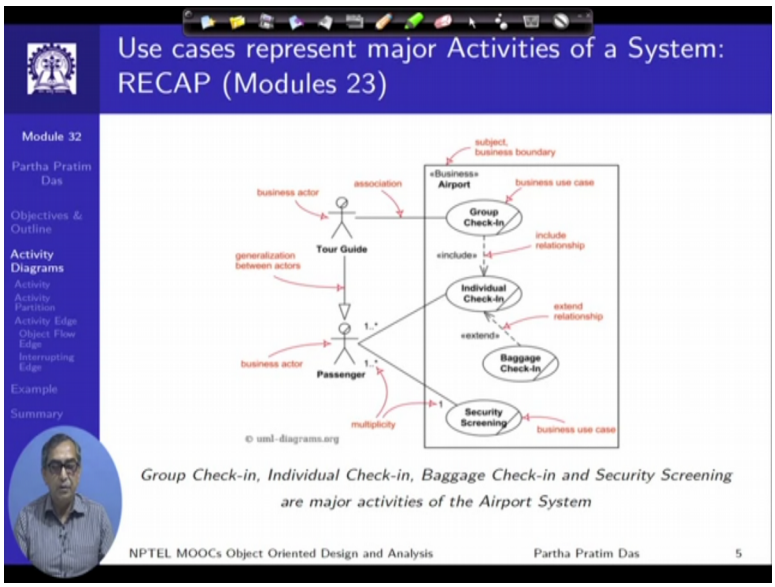
Since this will spread the 3 modules, so this will be the overall outline to understand activity diagrams and in every module superficially we will show those items in blue which we discuss in that module. So here these are the first 3 items that we will focus on and this outline would be available on the left of your screen all through the slides.

(Refer Slide Time: 01:09)



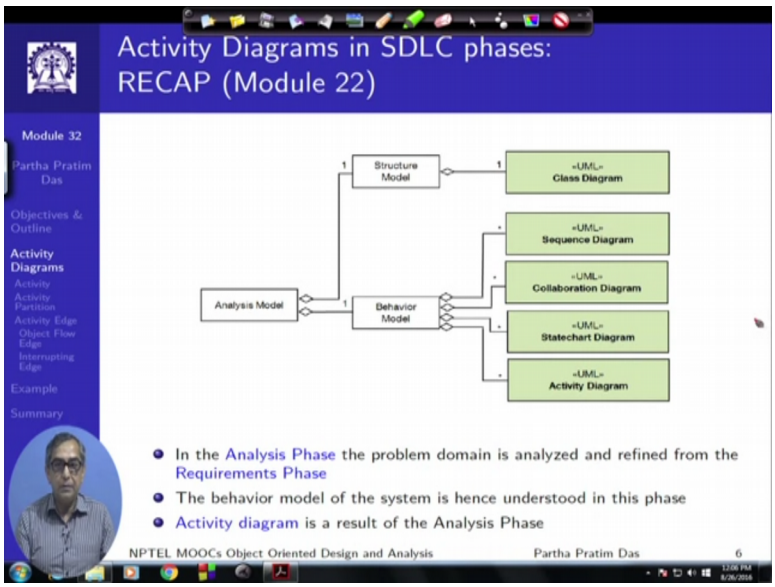
Now I will quickly point you to the earlier items that you should be certainly familiar with and now be master of, the client-server computing model which we have been regularly referring to. Activity diagram will have a deeper bearing with the realization of thee client-server computing model.

(Refer Slide Time: 01:34)



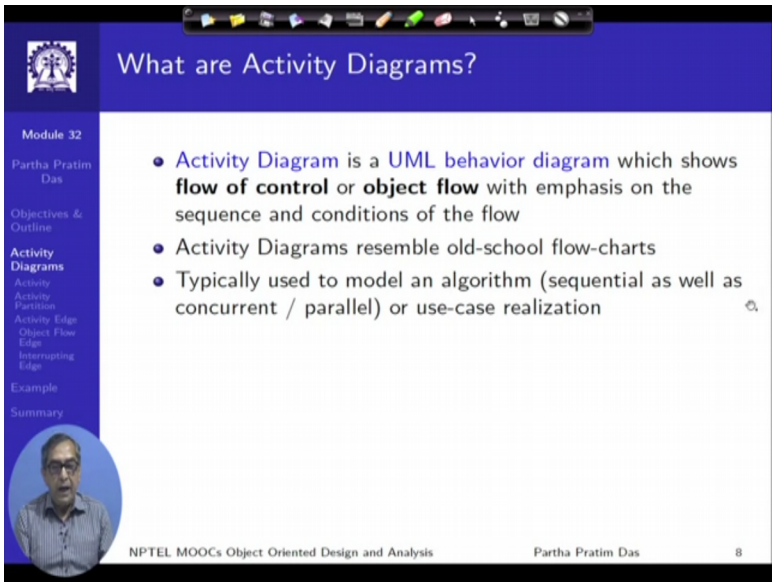
We have say a major reference to the use case diagram and the use cases because the use cases represent the major activities that happen for a system.

(Refer Slide Time: 01:49)



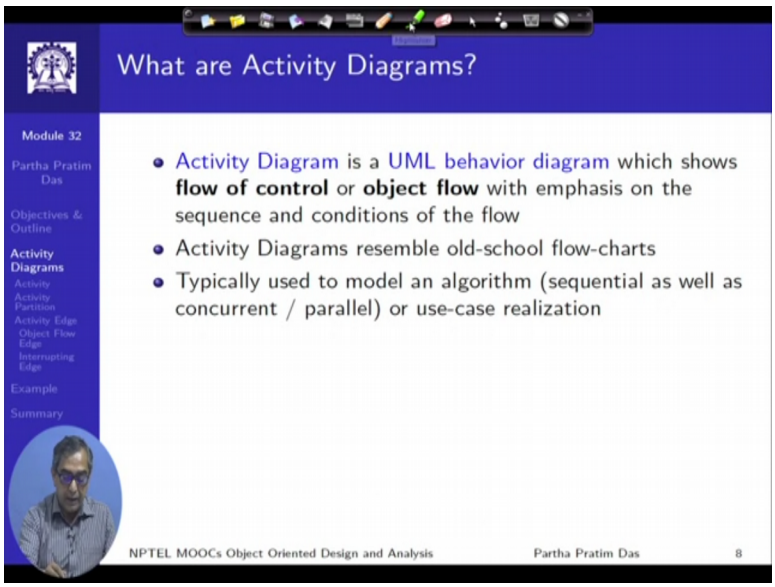
And for in terms of the sdlc phase this is where the activity diagram lies, so we are talking about primarily the analysis phase after the requirements phase where the basic understand has been made and here where the activities will have to be captured in terms of the activity diagram and that will continue through into the design phase where it is a part of the behavioral model where it is further refined based on the better understanding or the basic approach to the design.

(Refer Slide Time: 02:20)



So that is a the context in which we will discuss the activity diagrams and I will urge you that the previous diagrams that we have already discussed particularly the use case class and sequence diagram which have a frequent bearing on the activity diagrams particularly the use case and the sequence diagram. So you should clarify your understanding on those diagrams before you go through this video.

(Refer Slide Time: 03:00)



The activity diagram is a behavioral diagram am sorry, activity diagram is a uml behavioral diagram so it shows the flow of control or the object flow emphasizing the sequence and the conditions of flow that is the basic point. So now you will find that a quite a part of the activity diagram would look like what earlier we used to call the flow chart. So, they are kind of refinements or advancements on the flowchart representation which we used eeh a very early stage as a model to represent certain logic.

So, they are typically used to model algorithms, so so far, we have only talked about the objects the intended use cases, the properties and operations and in terms of the operation we have just talked about identifying the operation or in terms of use cases, we have just talked about identifying the use case. We know there is a request leave, we know there is a approve leave and so on, these are use cases and then they made their inroads into the corresponding class diagram.

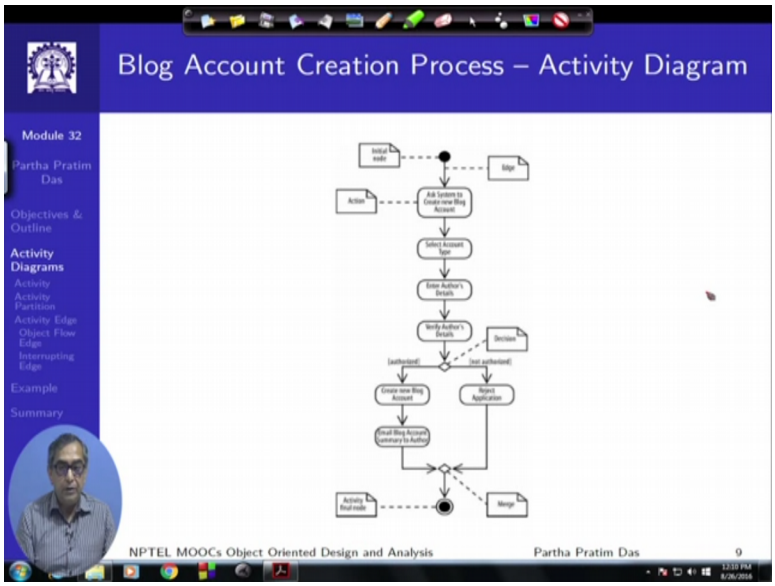
But here we would be more interested to really look into the algorithm to go in for performing the request leave, for performing the approved leave and so on. And importantly in today's system many at many a times, the algorithms are not necessarily sequential, they may be concurrent algorithms or they may be parallel algorithms. The basic difference being that a concurrent algorithm is where multiple threads of control apparently execute at the same time while I may actually have only a single, even a single processor.

So, concurrency basically talks about multiple things happening at the same time, we have talked about concurrency earlier too and in fact we could have multiple cores in the target system or multiple processors. So, the concurrent tasks could actually happen in a parallel mode so that they are not time sharing, they actually have separate executions possible. So, all of these should be

representable in terms of the activity diagram and so we will use all that to realize that use cases.

So that is the basic you know requirement of the activity diagram, that's the basic space in terms of the uml representation that we are trying to fil in.

(Refer Slide Time: 05:40)



So just before I go into the (E.g.: 05:42) I just put up a simple activity diagram. This is like a describing a process to create an account for blogging. So, these are the main things you can see that there is an initial node where everything starts, we say is a start node where everything is a starting for that process and then you see a series of boxes which describe the different steps that you do starting from their access, ask system to create new blog account.

Certainly, that is the first thing we will have to do, you will have to select what type of blogging account you want, then you enter the details that is entering authors details and then the author detail will have to be verified in terms of the email access and you know other required items, if payment is involved and all that. So, these are the different steps, these are different activities that this whole process will involve and then in some cases like in here.

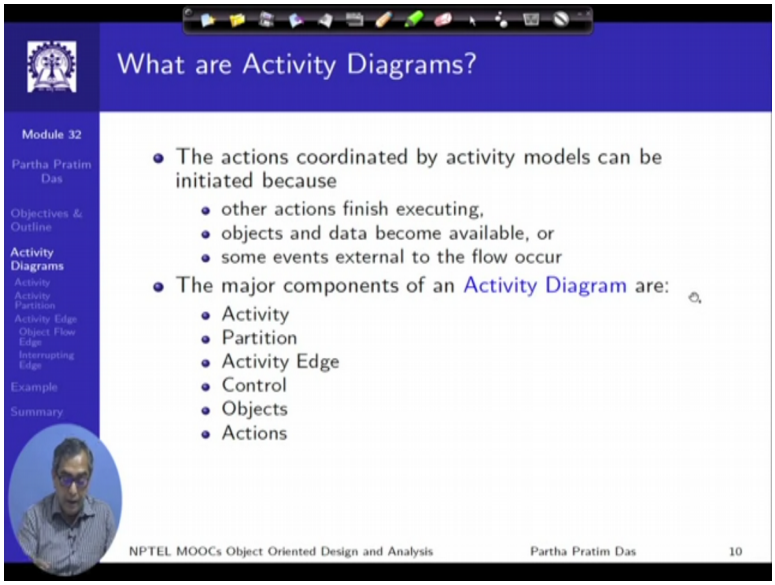
The outcome of verifying author details could be that we have an authorization that the verification turns out to be ok or the verification fails. So, based on that we have a branching in terms of the flow so it could be authorized or it could be non-authorized. If it is authorized, then you have further actions of creating the new blog account, email the blog account summary to the author and so on. On the other hand, if it is not authorized then the application gets rejected.

And based on either of these you know flow paths being taken, finally we reach a a common merge

point where which in this case denote the completion of this particular process in terms of an end node, final node. So, if you look into this I have taken particularly taken an example which looks pretty much like the flow chart, we will see that how we will have various different aspects so that it will get more enhanced from the flowchart.

But in this way activity diagram would be used to define describe different processes that correspond to the algorithms or correspond to the use case realization.

(Refer Slide Time: 08:00)

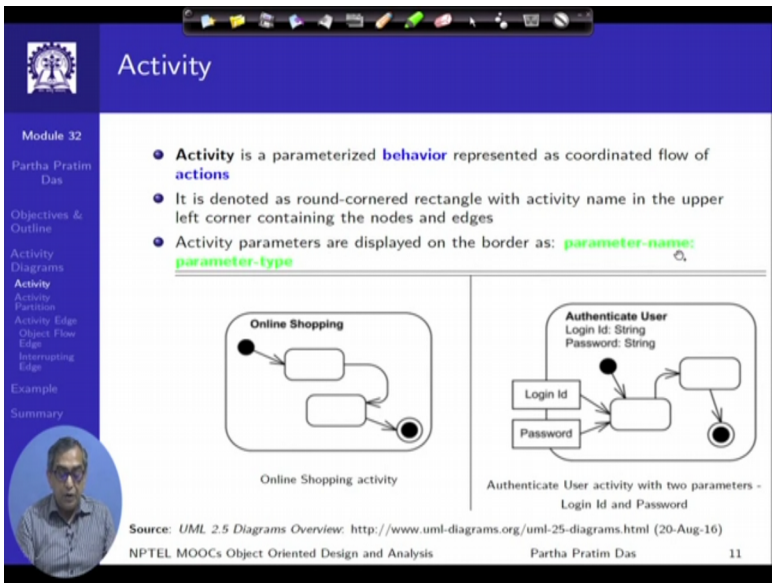


In terms of the so in formally speaking, the activity diagrams have actions which are coordinated by activity models that we are going to describe and these models can be initiated because some other action has finished. So, as we saw in the earlier, once you have filled in the author details then it goes for verification. So once one action finishes the other takes up. It may be initiated because certain object and data become available.

For example, you are logging in so you provide the user id and password. So, once you make that available then the action for verification of the login starts or may be because of some other events external to the flow is happening in terms of receiving a message token from someone or having certain system conditions and so on. So, activity diagram will show you all these actions and how they are interrelated. In terms of the components, these are the different components and I will not go through them right now.

We will take up these one by one and show how they are represented, what do they mean and how finally they help us achieve the activity description.

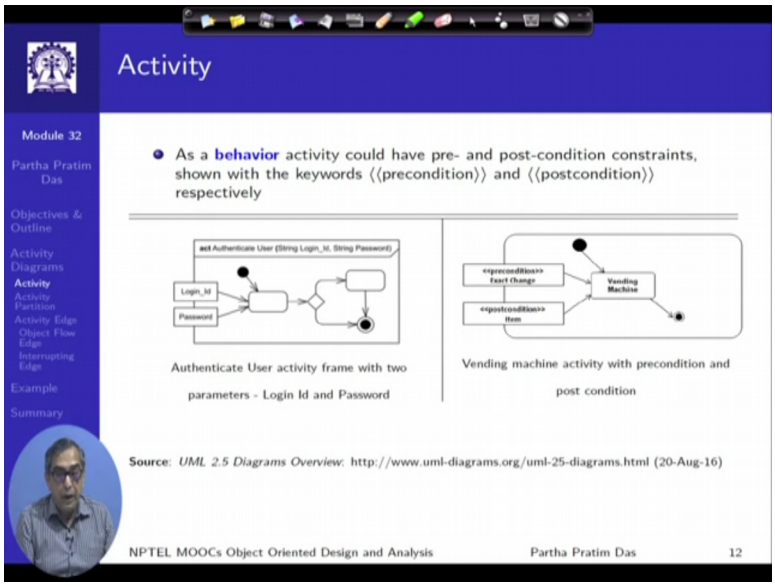
(Refer Slide Time: 09:19)



So certainly, the first is an activity which is certain kind of actions. So, here we have an online shopping activity which again in turn have certain sub activities in that. So online shopping these are not described here, we will come to the descriptions but it could that this has sub activities but what is usually unique in terms of activities that there is a start and there is an end. So, the object there is a unique point where the activity starts and there is a unique point where the activity ends.

It could be parameterized also, so this shows another activity where authenticate user activity which needs to take these 2 parameters login id and password does some sub activities and starting from the start node, which is the final node. Look at the particular type of rectangle that we show for describing activities, this is not a cornered rectangle that we typically use, this is kind of a rectangle with rounded border. So, when you see that you know that this is a activity there is a parameterized behavior of an activity that is being talked off.

(Refer Slide Time: 10:29)

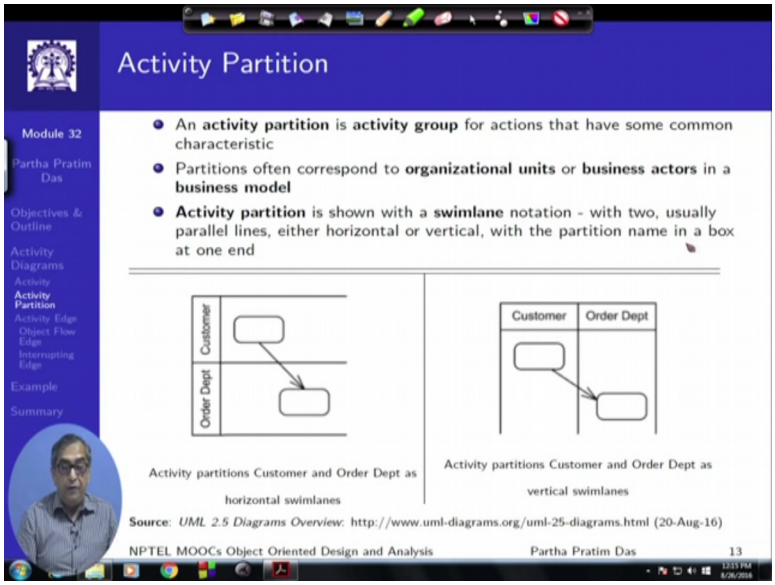


So, you could put a total set of activities, flow of activities in terms of a rectangular box also,

provide it a name as the activity action and provide the parameters for that. There could also be preconditions and post conditions on certain activities. For example, there is a vending machine, vending activity where the that is you that is a machine where you go put your coins and you can get a a drink or a new paper or something like that.

Certainly, the pre-condition is that you must be able to provide exact change in terms of point and the post condition is you actually receive item. So, this is the basic unit the basic activity that will build up activity diagram.

(Refer Slide Time: 11:20)

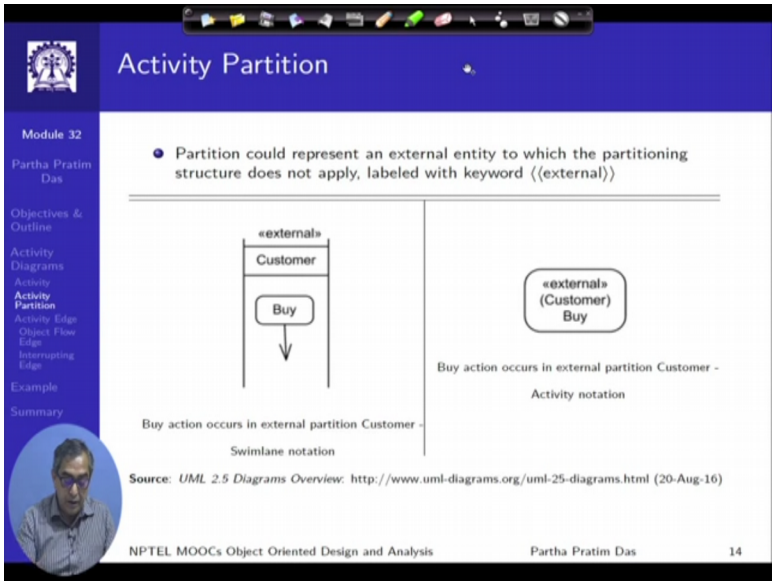


Often will find that there are different agents and typically if you relate back to the use case diagram these agents are actors and activities will relate to certain actors. So, when that happens then we try to partition the activity based on which actor is involved in this activity. So, it is a ordered department, there is a customer, so it could be in terms of actors or it could be in terms of other partitioning, it could be based on location.

It could be based on organizational structure and then you kind of draw swim lanes within which you show the activity so what it means is if I show this swim lane for order department then all activities that are shown in this partition shown in this swim lane are supposed to happen in the order department are supposed to be acted on by the order department here, it is for the customer and you can draw the swim lanes in a horizontal manner.

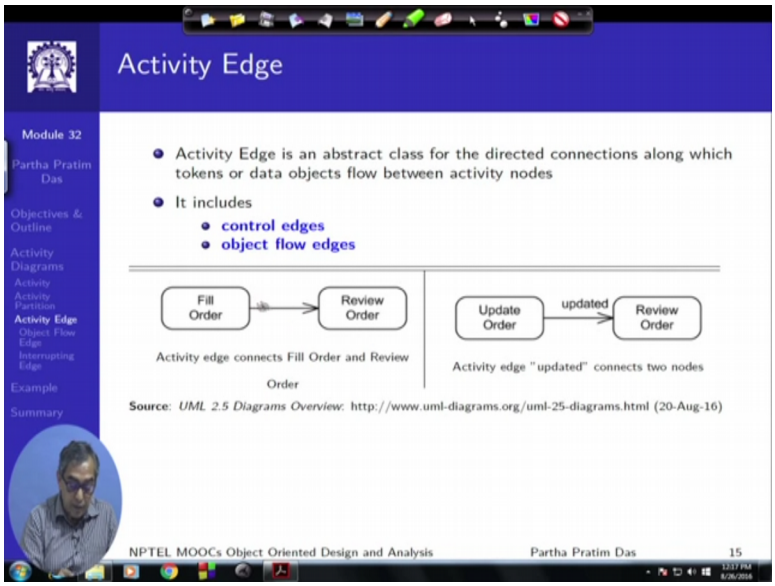
As you can see here these are horizontal swim lanes or you can draw them in a vertical manner, that clearly the swim lanes will clearly show you that how the activities can be organized in different parts of the system.

(Refer Slide Time: 12:33)



So, with the activities and partitions you could also have actually external partitions, external partition is one where it is not a part of the system that you are build in but some certain things that you expect the external agent ex actor to really act on and prove take part in the activity diagram as a whole.

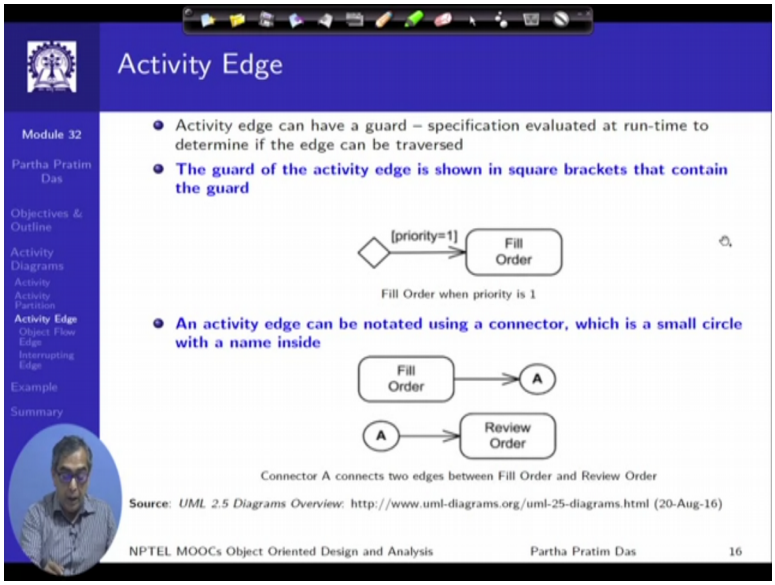
(Refer Slide Time: 13:02)



Activities have edges so there is activity fill order here, there is a activity review order here and there is a directed edge between them. Certainly, this shown that from the final node of this activity to the initial node of this activity there is a flow that is happening and we will see that typically this edge would be control edges or object flow edges where objects could also flow tokens could also flow from one activity to other through these edges.

They could the edges could have certain annotations which say that what is the effect of that edge and so on. Activity edges with the activity nodes really make up the diagram.

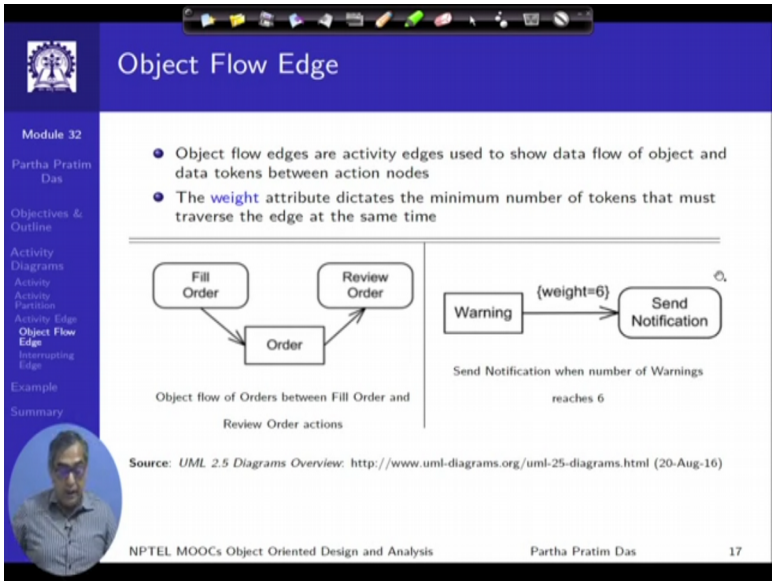
(Refer Slide Time: 13:49)



In further refinement, edges could have guards, we have been talking about the concept of guards in different context. So, what does this mean the we have not yet explained this, this is this basically is some kind of a decision or merge node, we will see later. But from this end of the activity you fill order activity the if the control has to flow in that then we are putting within square bracket a certain Boolean condition.

So, this means that this edge will be taken provided this condition turns out to be true. You could show the edges I mean in a complex diagram it may not be always possible to draw all the edges, it might just become a clutter of lines going from here and there. So, you could also you know divide a flow in terms of showing certain level markers. So, you say that the flow order this this a is just a continuity marker, so it goes up to a and wherever else you have a, you know that there is basically the same continuity. So, it you are showing this edge in 2 parts.

(Refer Slide Time: 14:56)

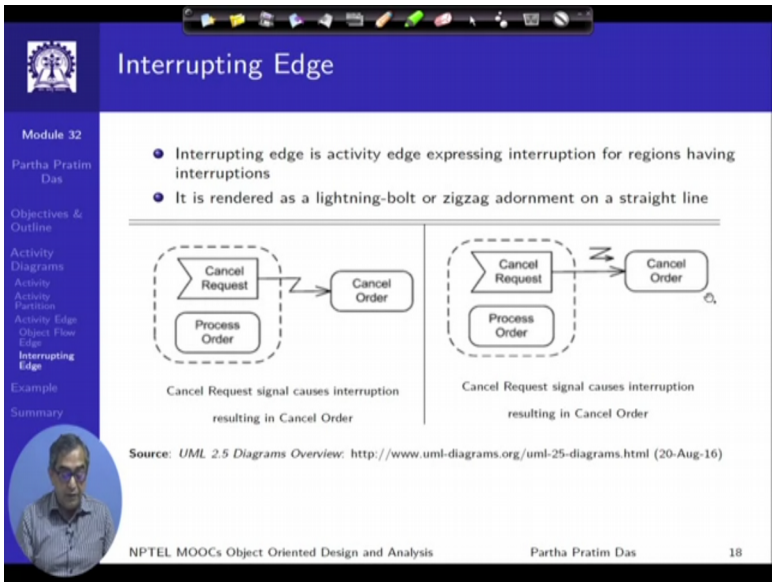


So, these are your activity edges, then there could be the activity edges could have objects that is this is how an activity, so there is an activity edge here as we saw and in that instead of directly showing the edge, we are showing an object. You understand that it is an object by the basic type of rectangle this is a regular cornered rectangle which means that actually as thee fill order goes to the review order as the this edge is taken, this particular order will move from fill order to the review order.

So, whenever you have objects identified which move from one activity to the other along an edge, you will say that there is an object flow edge. It could be certain cases objects could be implicit also for example here, we say that the warning object will flow to the send notification provided you have this condition, weight is 6 which would mean that there are 6 instances of this object so that the send notification will happen.

So, the situation that it will describe is if warning is happening, then for every warning you don't send an notification to the user but if there are 6 warnings that have accumulated then you send a notification. So you can model such things here, you just note that this is actually an object and this is an activity.

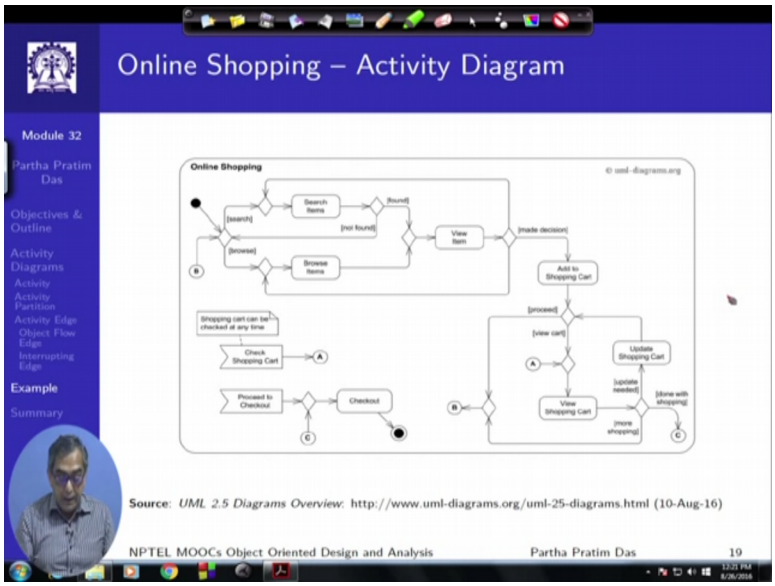
(Refer Slide Time: 16:26)



There could be other kinds of edges also. One important is thee interrupting edge, interrupting edge is activity edge expressing interruption or regions having interactions. For some reason, the normal designed flow designed action cannot happen, so you have an interaction. So, for example you have a cancel request here. If the cancel request is done then naturally the order needs to be cancelled so the normal flow will get disrupted.

So, this is how this through this zigzag, you show that this is a interrupting edge, you could have the edge itself zigzag or you could have a straight edge like in here and have a zigzag adornment symbol port on top of this edge. So, this will represent the interruption taking place in the activity diagram.

(Refer Slide Time: 17:14)



So, with this let us try to understand there are other items here as well in this diagram but let us just try to follow an online shopping activity diagram. So, this is where you start here we are based on 2 guards, we are making a decision, we will talk about decision more. So, you could what is happening if you want to do a online shopping, you can either search for an item or you can browse through the item through the list.

So, if you browse through then you have the browse item activity which is the process of browser browsing. If you choose to search then you have the search item activity which is a process of search and if you if you try to search then it is possible that you have found the item or you may not have found the item. So again, we have a decision where if you found then you proceed, if you are not if you have not found then you come back and possibly try to do the search again,

We will what these nodes would mean otherwise if we have you could have either searched and found an item or you have browsed and come to an item so then you go into the details of looking at the item and in that process, you might decide that you would like to buy some item where you have made the decision you proceed further and if you not then you would basically go back in terms of either continuing on the search or continuing on thee browse.

If you have made the decision then you do the next activity which is adding to the cart where as you

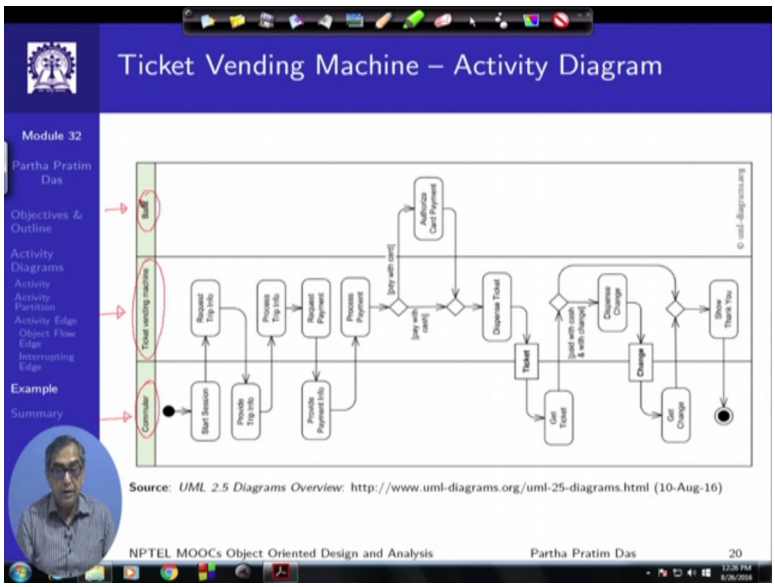
add to the cart, you could be then you could view the cart and view this and then go forward or it could come to this whole process of view the cart from a completely different activity which is shopping cart can be checked at any time. So here as you are looking for searching or browsing for an item similarly you could just go an look at the status of the shopping cart seeing what are the items that you had selected before.

So, the check shopping cart item takes you to the next activity which is labeled at A which means that it actually comes here so you could either come through this process of search or browse, making a decision, adding to the cart and then viewing the cart or you could directly come from here viewing what you had seen earlier in terms of the cart and then either of the cases you actually move to viewing the shopping cart and after you have viewed you could select to add

And update the shopping cart or you could just decide to continue, do more shopping so you go back to B, just find out where is B, the B is here. So, you have added that item and now you come back and you just do not want to immediately go for checkout so you want to come back and start searching and browsing again. Otherwise it is possible that you said okay I am done with shopping I have no more items to choose and I put to the cart but I would like to check out pay for and actually buy.

So, you go to this label C which is what you find here, so which is the check-out process, you could come to checkout directly also. So, this is the kind of flow so here I have taken an example where you primarily see the flowchart kind of activity is only we will soon see the other kinds and you can find out how in a very compact, nice as swell as expressive form, you can represent the whole set of actions that you would like to do in terms of online shopping activity.

(Refer Slide Time: 21:07)



This is another example I am sorry about the orientation of this diagram because it is a in putting vertically it becomes too small and what am trying to show are basically the activity partitions so you can see there are 3 swim lanes here. The commuter, the ticket vending machine and the bank. So, it basically it's about as if you have gone to a rail station or a bus station and there is a vending machine into which you can say why you want to go

And then put appropriate money through card or cash and then the ticket is issued so it starts up this is your certainly the start node, you start the session naturally that is, so if you start here, it is done by the computer. So, you have the start session activity and that certainly will request for a trip information as to I want to go from plus x to plus y, I want to go from Hyderabad to say Mumbai and so on. Then so that goes to a different swim lane of ticket vending machine.

Now this activity will result in the providing the trip information which is again for the commuter to look at. So, you can you can see in this bunch of activities here, the edges keep on crossing back and forth between the 2 swim lanes because it shows an interaction between the commuter and thee ticket vending machine happening one after the other. So, after the trip info has been provided, you can confirm that I want to do this trip so you send it back and send a go into the next activity which is processing trip info.

Then the ticket vending machine will request for payment which again comes back to you for providing prevent information goes back to ticket vending machine for processing payment and you have a choice to pay with card or with cash. If it is with card then it goes to a further activity which is now in the third swim lane of bank which has to decide the validity and all those. So, I am not showing all those details and if it is with cash then you directly go in.

Then the once the payment has been met then the ticket vending machine will dispense the ticket that means it will issue that ticket, then this is this you can see is a object flow edge. So now so fat it was just a information now an actual object has to come out which is a ticket. So, the dispense ticket activity goes to the ticket activity and that brings in an object here. So, dispense tickets as generated an object which flows in to the gate to gate activity.

And so, you have pa paid with then there are subsequent actions like if you have paid with cash then there may be some change that you are supposed to get. So, if you are supposed to get changed then that activity of dispensing change will happen and the change as an object again will come back to the gate change activity but if you are paid exact then this is not required. So, you bypass and come directly here which is the next activity which is basically termination ba the machine says that thank you for buying the ticket and so on, blah blah blah.

And you end up with the final node which is you have ended this ticket vending process. So, this kind of a process like we clearly show here that it is it's still a flowchart in the sense that it is giving us a one single sequential algorithm for doing the whole thing but representing in this kind of swim lanes, this will clearly tell us that who are the responsible persons the actors that are involved in different activities and it will also tell us.

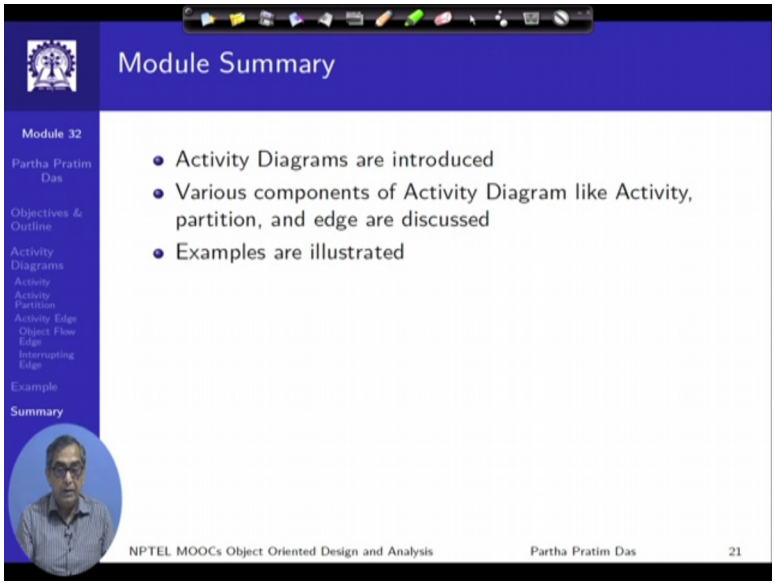
So, you will find that you will be able to get to this swim lanes very easily if you look into how the use cases were looking like. So, these are different parts of the use cases that you had in this corresponding ticket vending machine if you build that use case diagram and these are the different actors and you accumulate the activities under the respective swim lanes so you know that in terms of the commuter, what are the things that need to be done.

Commuter possibly is more more a manual person, more a person whereas in terms of the ticket vending machine, all the activities that you have here are parts of the code system that you are trying to build in. so you will have to really design deeply through that and in terms of the bank swim lane, possibly bank will have their own authorization processes. So, it will already be existing so all that you need to really think about are the edges that happen with them that is how you send the information to the bank for authentication, what the bank sends out in return and so on.

Of courses, there could be more detailing of this activity further down if you look into from the bank's point of view. So, the swim lanes really help you isolate as to what activity should be where

and what you should primarily be designing for and what are the activities which will have will need interfaces with the external world, so that you just primarily you need to work on the protocols and make assumptions in terms of how those activity edges work with the system.

(Refer Slide Time: 26:37)



So, to summarize in this we have introduced in this module, we have introduced the activity diagram and as I said this is a first of the 3-part discussion on activity diagram. We have talked about the basic components of activity diagram which is the activity partitions and the edges particularly including the object flow and interruption edges and we have explained through some illustrative examples which will continue in the next modules as well.