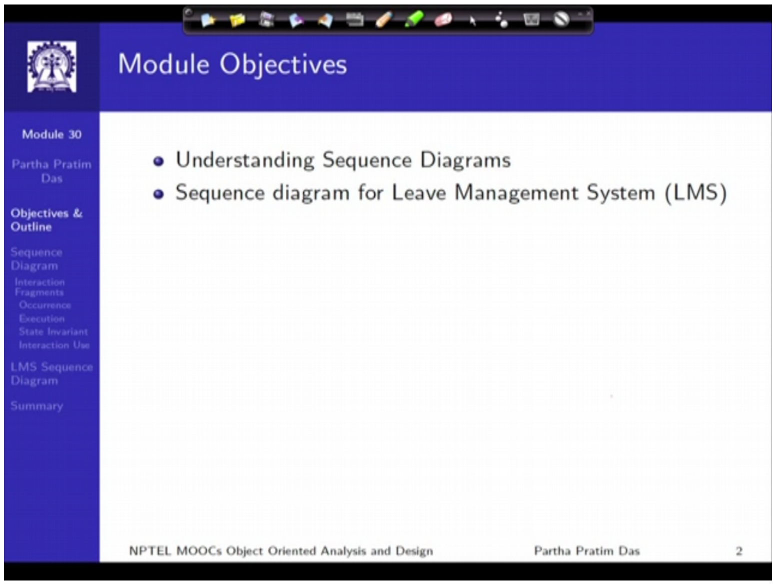
Object-Oriented Analysis and Design

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

Lecture – 42
Sequence Diagrams Part 2

Welcome to module 30 of object-oriented analysis and design. We have been discussing about sequence diagrams.

(Refer Slide Time: 00:29)



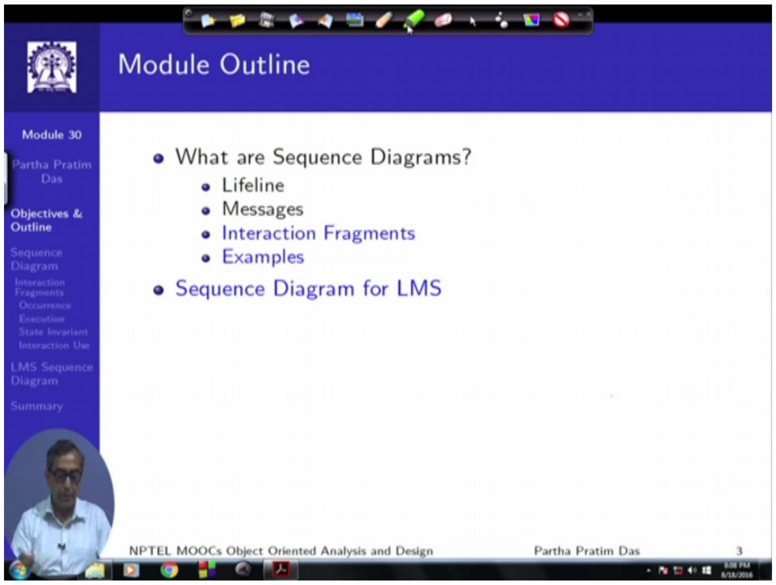
Module Objectives

- Understanding Sequence Diagrams
- Sequence diagram for Leave Management System (LMS)

NPTEL MOOCs Object Oriented Analysis and Design — Partha Pratim Das — 2

So, we will continue that discussion complete understanding of that sequence diagram and we will take a brief look into building the sequence diagram of our lms exercise.

(Refer Slide Time: 00:44)



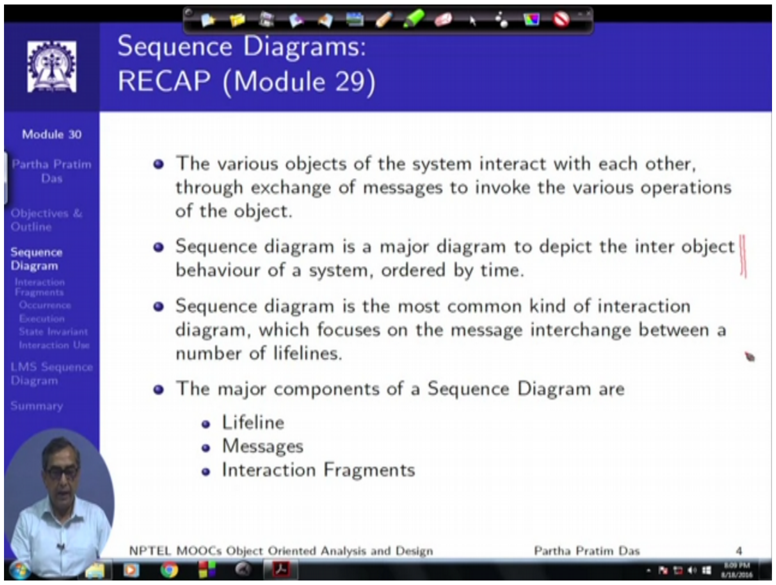
Module Outline

- What are Sequence Diagrams?
  - Lifeline
  - Messages
  - Interaction Fragments
  - Examples
- Sequence Diagram for LMS

NPTEL MOOCs Object Oriented Analysis and Design — Partha Pratim Das — 3

So, in terms of the sequence diagram we have already looked at the basic concepts that of a lifeline which is how does a object exist in the system over time, how it can be created within this diagram, how it can be deleted and how it can exchange messages and in that context, we have seen various types of messages synchronous, asynchronous, create, delete and reply messages and we have seen their classification by response type as well. We will continue in this module to introduce the concept of interaction fragments and look at some more examples.

(Refer Slide Time: 01:33)



Sequence Diagrams: RECAP (Module 29)

Module 30

Partha Pratim Das

Objectives & Outline

Sequence Diagram

Interaction Fragments

Occurrence

Execution

State Invariant

Interaction Use

LMS Sequence Diagram

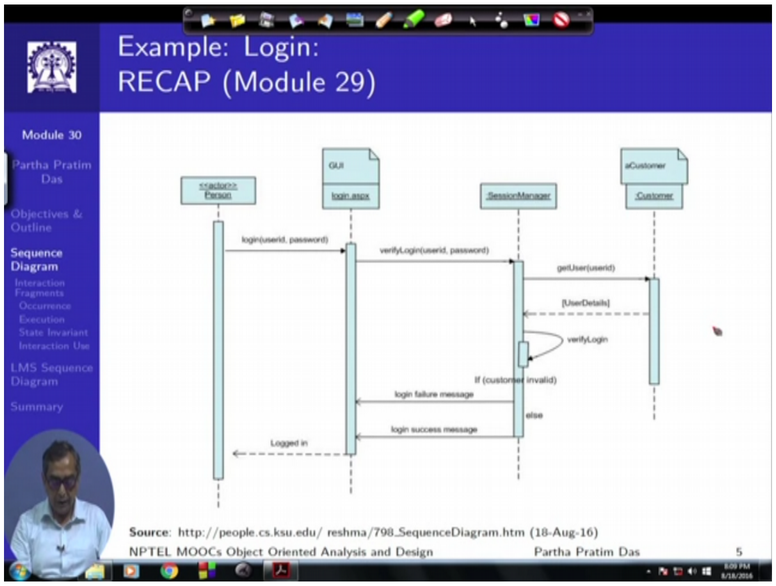
Summary

- The various objects of the system interact with each other, through exchange of messages to invoke the various operations of the object.
- Sequence diagram is a major diagram to depict the inter object behaviour of a system, ordered by time.
- Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a number of lifelines.
- The major components of a Sequence Diagram are
  - Lifeline
  - Messages
  - Interaction Fragments

NPTEL MOOCs Object Oriented Analysis and Design Partha Pratim Das 4

So, this is what we have already seen that in the sequence diagram is a major diagram which depicts the inter object behavior and it is a kind of behavioral interaction diagram and these are the main components. So, this is just to bring you back to the same point.

(Refer Slide Time: 01:56)



Example: Login: RECAP (Module 29)

Module 30

Partha Pratim Das

Objectives & Outline

Sequence Diagram

Interaction Fragments

Occurrence

Execution

State Invariant

Interaction Use

LMS Sequence Diagram

Summary

Source: http://people.cs.ksu.edu/ reshma/798_SequenceDiagram.htm (18-Aug-16)

NPTEL MOOCs Object Oriented Analysis and Design Partha Pratim Das 5

Now this is one of the examples that we had talked of. So am sure you have already revised this. Eh this we have left as an exercise last time. So, this is like a sequence diagram for a place order kind of use case in an online store. So, if we take a closer look this time, we have a person object o who is basically initiating this order activity. So, there is a gui to choose the product, so naturally the person sends a select message to the gui for the selection.

You can look at here, this is a synchronous message. Once this message has been received then in turn the gui will send a message, a product, a product that has been selected, that message to the session manager. The session manager in turn will add it to the will try to add it to the shopping cart. So, it is like you have selected the item, you go and say that add to cart. So, once you have selected before it can be added to cart, naturally the system needs to check in the session manager in terms of what is the present status of that item.

Whether the item is available, whether the item has a so what is the current price of that item and so on. So, the session manager sends the message add cart item to the shopping cart and once the shopping cart gets that so what shopping ui here shopping cart has received the add cart item, then the shopping cart has to actually create a new cart item. So, it does not knew because this cart item was not there, it will have to be created and put in the cart, so of the cart item class, a new cart item is created.

And that cart item creation once this is over then a reply response message goes back to the shopping cart who then invokes the add cart item on itself. So now the cart item object has been created, so now it needs to be added to the cart. So now it is a self-message, once that has been added then you calculate the price of the resultant price because of this addition. So that is also a self-message and the item has been added, it is ready for ordering now.

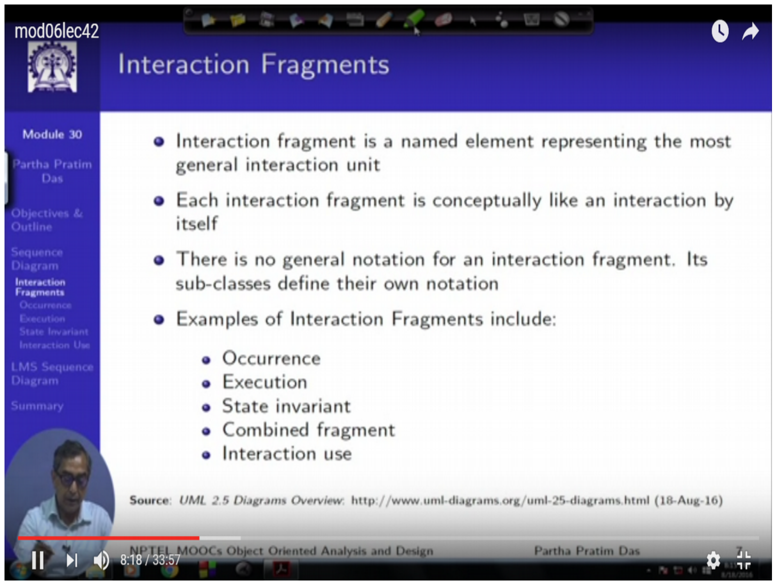
Now in this way the person can actually add several items to the cart and once she is ready for a checkout that is checkout as know is a process by which we say that the items that are put in the cart you confirm that you want to buy them and therefore are ready to make the payment. So, you send the checkout message to the cart and in that checkout message to the cart, then the cart in turn has to update the profile of the customer because it has to check whether you are whether you had earlier orders.

Whether you have some default payment terms, whether you have already have some credit with this store and so on. So, it will send a message update profile to the customer instance and get a reply from that. Once that has been update then the shopping cart again sends a calculate total price, calc total price message to itself so that all items that have been added to the cart, their total price will be computed and on completion of that, a new message place order is initiated to place the order on the order object.

And once that place order execution is over; the order confirmation notification is sent back. Naturally this is just an overall flow of how the order is placed, of course there are several details for example there are several failure situations that finally will need to be modeled for example when we are trying to add a add an item ca cart item and calculate price, it is it is quite possible that actually the product that was selected, that item may not be available to be added to the cart.

So, what will happen in that case will have to check for that. When we update the profile of the customer here, it is possible that the customer has a profile where the sale order may not be a valid one. Certainly, there is a whole issue of a payment which is not shown here and it is quite possible that the placement of order may not go through in that case, we will not have a order confirmation but we will have a order regret kind of notification.

But leaving that aside that if we just work on the kind of as we say the happy paths as to what is likely expected path of flow behavior then this is kind of a diagram which show how place order will work. So, a task often in terms of sequence diagram would be to actually think about the total flow of events in the system in terms of the classes, the named entities and identify what message need to go from which object to what object so that in that order in that sequence we actually achieve the targeted requirements.
(Refer Slide Time: 08:16)


Interaction Fragments

- Interaction fragment is a named element representing the most general interaction unit
- Each interaction fragment is conceptually like an interaction by itself
- There is no general notation for an interaction fragment. Its sub-classes define their own notation
- Examples of Interaction Fragments include:
  - Occurrence
  - Execution
  - State invariant
  - Combined fragment
  - Interaction use

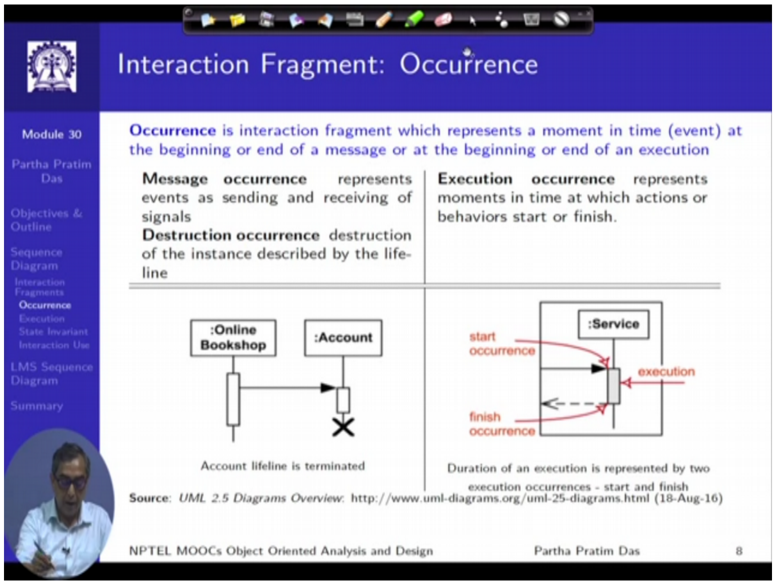
Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Now let us discuss about the interaction fragments. Interaction fragment is a named element representing the most general interaction unit. So, it is like the main idea of interaction fragments a the sequence diagram as a whole is showing you several different interactions a every there is a unit interaction between one class with another as one message is send and there is a sequence of messages sequence of synchronous, asynchronous calls, creation, deletions, all these giving you total complex picture of interaction.

But if we just take a part of it then that part itself could be something meaningful and if we see that that part needs to be performed several times in several different places in sequence diagram or in several different sequence diagrams then we can actually take those out and kind of make them as a as a reusable interaction part. So, interaction fragments have the general objective of talking about interactions at different levels which can be taken out and it reused at different places can be referred at different places and so on.

As such there is no general notation for interaction fragments there are different kinds of fragments and each fragment has some implicit notation of its own. Eh these are the common types of fragments we will now take a look into what these interaction fragments mean.

(Refer Slide Time: 10:03)



Interaction Fragment: Occurrence

Occurrence is interaction fragment which represents a moment in time (event) at the beginning or end of a message or at the beginning or end of an execution

Message occurrence represents events as sending and receiving of signals

Destruction occurrence destruction of the instance described by the life-line

Execution occurrence represents moments in time at which actions or behaviors start or finish.

Account lifeline is terminated

Duration of an execution is represented by two execution occurrences - start and finish

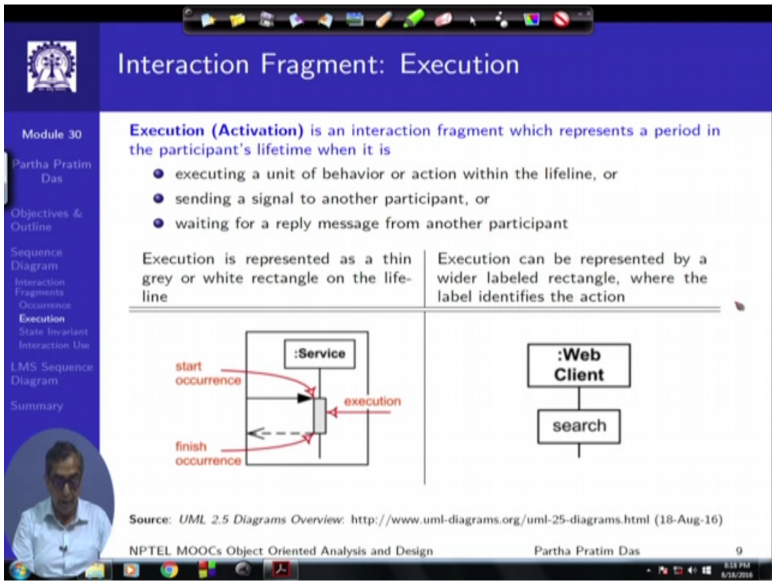
Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

NPTEL MOOCs Object Oriented Analysis and Design Partha Pratim Das 8

The first is the interaction fragment of occurrence. The occurrence is an interaction fragment which represents the moment in time or event at the beginning or end of a message or at the beginning or end of execution. So, if a message is sent from say online book store shop to account then this message has an event at the beginning, it has an event at the end. So, these are the occurrence specifications, similarly an execution is happening so there is a beginning of the execution and there is a end of the execution.

These are called the occurrences, so those occurrence specifications can be used to particularly associate different conditions different constraints at those. So, we typically have message occurrence which relates to the events at the sending and receiving signals, we have execution occurrence which is in terms of the beginning and end of execution and we may have a destruction occurrence as well which relates to whenever the objects are destroyed. So, this is the first kind of interaction fragment that is frequently used in the sequence diagram.

(Refer Slide Time: 11:27)



Interaction Fragment: Execution

Execution (Activation) is an interaction fragment which represents a period in the participant's lifetime when it is

- executing a unit of behavior or action within the lifeline, or
- sending a signal to another participant, or
- waiting for a reply message from another participant

Execution is represented as a thin grey or white rectangle on the lifeline	Execution can be represented by a wider labeled rectangle, where the label identifies the action

Source: *UML 2.5 Diagrams Overview.* http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

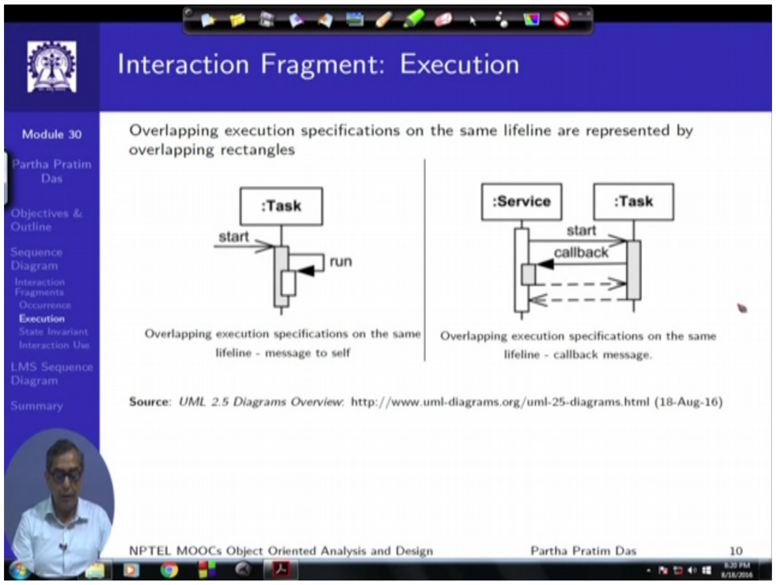
NPTEL MOOCs Object Oriented Analysis and Design Partha Pratim Das 9

The second is an execution fragment, it is often called the activation fragment also. In an interaction, it is an interaction fragment which represents a period of time in the participants lifeline when it is actually executing. So execu it is executing a unit behavior action, sending signal to another participant or waiting for reply message from another participant. So, a an execution like in here hav has started based on some message.

And then during this execution one part is it is executing so that is the one sequence of things that can happen. Besides it may need to send some signal to some other participant, message to some other participant. If it is synchronous then it will have to wait for receiving the response on the signal that has been sent. So wee depict these in terms of the execution fragment, so the execution is typically drawn out in terms of rectangular boxes, we have been saying this already so you have understood.

They are filled in grey or they could be just left blank or alternately we could also use a bigger kind of a rectangle and write the name of the execution so here search is an execution fragment. So, this is an execution fragment so that basically means that it it is kind of an interaction, it is a kind of a small piece of interaction which happens at different times in the sequence diagram and builds up the total sequence of activities and we designate that with the start occurrence and end occurrence and the actual execution and signal link.

(Refer Slide Time: 13:28)


Interaction Fragment: Execution

Module 30

Partha Pratim Das

Objectives & Outline

Sequence Diagram

Interaction Fragments

Occurrence

Execution

State Invariant

Interaction Use

LMS Sequence Diagram

Summary

Overlapping execution specifications on the same lifeline are represented by overlapping rectangles

Overlapping execution specifications on the same lifeline - message to self

Overlapping execution specifications on the same lifeline - callback message.

Source: *UML 2.5 Diagrams Overview*. http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

NPTEL MOOCs Object Oriented Analysis and Design Partha Pratim Das 10

Interestingly it is possible that the interaction fragment of execution could overlap. For example, here we see so here you can see one in grey and then another in brown. So basically, here the task has had started an execution and at within this the task finds that it needs to send a message for getting some service. It so happens that this object the task object itself is a provider of the service. So, it sends a message back to itself, if I sends a message back to itself then in that same timeline it is actually doing another execution.

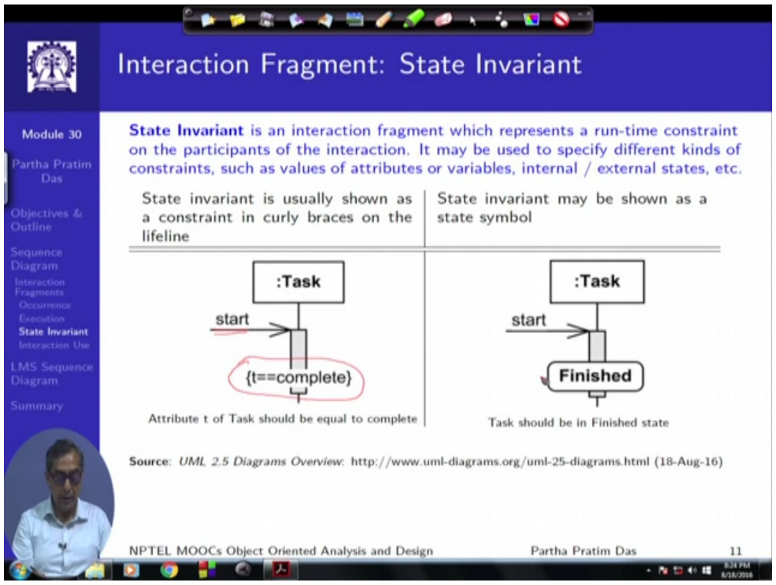
So that is what is shown by the overlap. So, when you have overlap, that means basically the same object is having multiple related executions going on at the same time. Here this is a synchronous one so when the grey one, the kind of parent one sends the run message to the to start the execution of the white part, this part then naturally the grey part goes on to fault. Because this is on a synchronous call. It can also be a different way it can overlap is if we have some situation of a call back.

So, call back is like this here you have seen that the basically the message is back to itself. But it could also be that the message has been sent from the service to the task, the start messages it is an asynchronous one, so the start is continuing to execute. The task in after doing some work in the execution find that it need some service it needs some execution to be done by the service itself. So, it sends a call back message. Why is it send call back? Because service had started the task execution and task is referring it back to the service.

So, it is kind of a loop finished. Here the loop is on the cell, here the loop is through another object. Now when it does that since this was asynchronous, the service still responsive, the service is executing other ways it would have been on hold and then those those would have been nobody to receive this call but message so as a call back is received, a new execution overlapped on starts. And since this is synchronous at this point the execution of the task is on hold but the execution has fresh execution has started in service overlapped with earlier execution.

And when that is over you send the response back, when the response goes back again the task execution continues and finally when the task execution ends then you send the reply back. So, there are 2 situations as we had seen where overlapped execution specifications are possible one where there is a self-message and one where there is a call back.

(Refer Slide Time: 16:49)



Interaction Fragment: State Invariant

State Invariant is an interaction fragment which represents a run-time constraint on the participants of the interaction. It may be used to specify different kinds of constraints, such as values of attributes or variables, internal / external states, etc.

State invariant is usually shown as a constraint in curly braces on the lifeline	State invariant may be shown as a state symbol

Source: *UML 2.5 Diagrams Overview*. http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

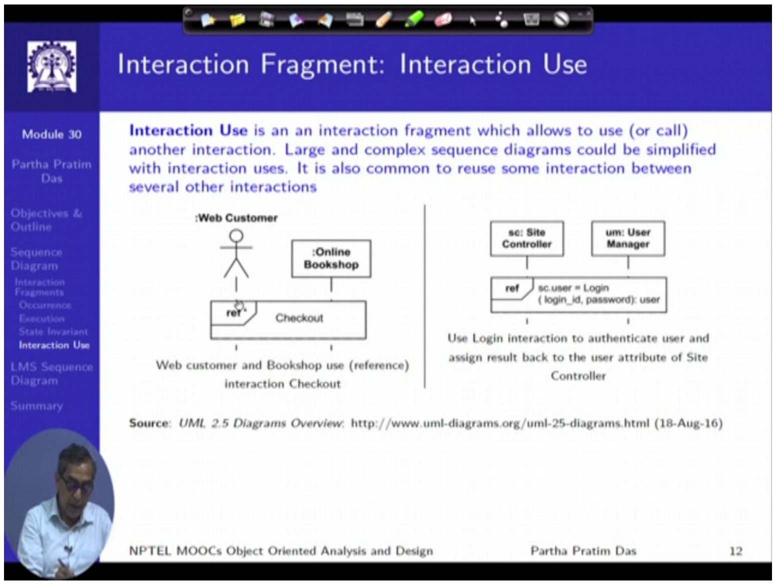
NPTEL MOOCs Object Oriented Analysis and Design Partha Pratim Das 11

There are few other interaction fragments which are used, one is called the state variant which basically says if you look at this, those execution which started here with this start message and here we write within curly braces that t is equal to complete. So, which means that there is an attribute t in the task and we want to say that when there is a execution will end, then t value of t should be complete. So that is what is known as a state invariant so we are saying that whatever may happen.

Whatever parameters that the start message could have but I need that whenever the execution ends, t is equal to complete. So, the state that it reaches must be invariant. So, it could be either it in terms of this kind of a condition statement or it could be written in terms of actual state values. So here what we are saying that when this execution ends, then the state must be finished. So that is that is the name of the state, it could be any any other state also.

So, this is known as a kind of state invariant so you as you can see this are more like constraints that we can put to ensure that the correct state has been raised or correct situation has been raised in terms of the sequence computation.

(Refer Slide Time: 18:25)



Interaction Fragment: Interaction Use

Module 30

Partha Pratim Das

Objectives & Outline

Sequence Diagram

Interaction Fragments

Occurrence

Execution

State Invariant

Interaction Use

LMS Sequence Diagram

Summary

Interaction Use is an an interaction fragment which allows to use (or call) another interaction. Large and complex sequence diagrams could be simplified with interaction uses. It is also common to reuse some interaction between several other interactions

Web customer and Bookshop use (reference) interaction Checkout

Use Login interaction to authenticate user and assign result back to the user attribute of Site Controller

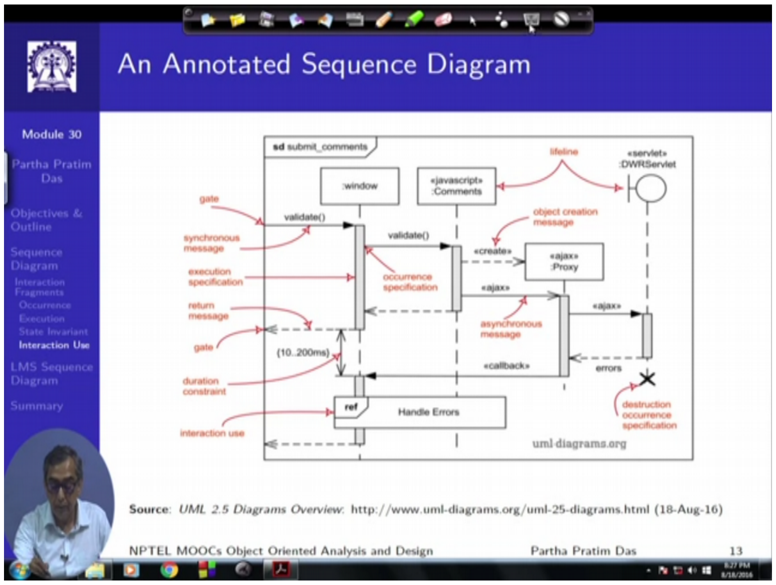
Source: *UML 2.5 Diagrams Overview.* http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

NPTEL MOOCs Object Oriented Analysis and Design Partha Pratim Das 12

Finally, there is an interaction called interaction use which is kind of a sub routine we can say. So, these are so if you have defined a an interaction somewhere say the checkout interaction that we just observe and then we can say that that checkout interaction part I can refer here which is ref means that I can actually give a name to that set of interactions, the sequence of messages and the execution and all that and at a different place I can just refer to it without actually again including all the details of this checkout interaction.

We could also instead of just doing a flat include, we could also actually pass messages, the login and password messages. So, login assuming that login was earlier specified as a interaction fragment, I can refer to that passing the new parameters here I do not need to detail out the whole of what this interaction is. When you do that we say this is an interaction use.

(Refer Slide Time: 19:43)



An Annotated Sequence Diagram

Module 30

Partha Pratim Das

Objectives & Outline

Sequence Diagram

Interaction Fragments

Occurrence

Execution

State Invariant

Interaction Use

LMS Sequence Diagram

Summary

Source: *UML 2.5 Diagrams Overview*. http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

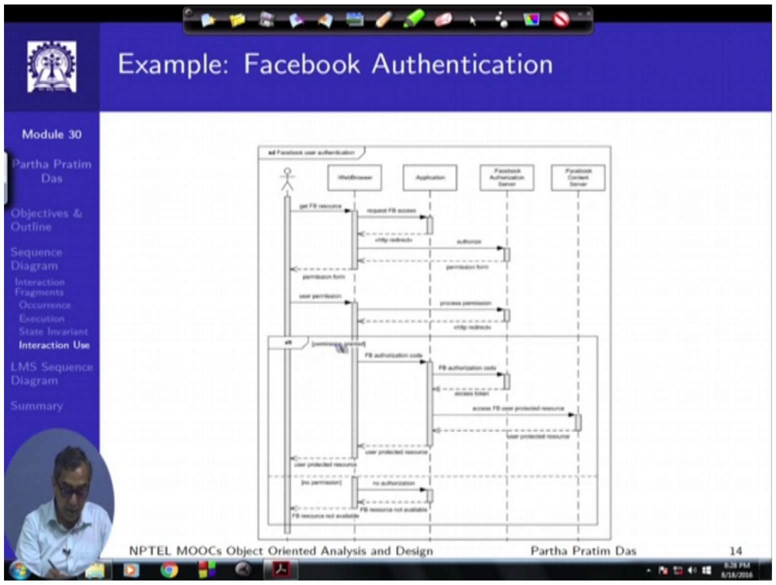
NPTEL MOOCs Object Oriented Analysis and Design Partha Pratim Das 13

So, these are the different kinds of interaction fragments that are that are commonly available for. So, if you look into this annotated diagram, you can see different places where this interaction fragments are can be seen. For example, this is occurrence specification, there will be an occurrence specification as well. There will be in terms of the interaction there is a interaction use which is trying to use the handle error which has been defined suppose at least somewhere else.

Then there is a separate occurrence specification for destruction. So, you have now you have already understood the different lifeline and message details, you are understating the interaction fragments also. So now you are more or less equipped to understand the whole diagram and for example this is a there is a execution fragment with the execution specification, there is a start here, there is a end occurrence here and so on.

So, you can go through the we have been referring to this set of similar diagrams according to as we develop, as we learn more and more concepts in the sequence diagram so you can refer to them all here and get a better understanding.

(Refer Slide Time: 21:22)



Example: Facebook Authentication

Module 30

Partha Pratim Das

Objectives & Outline

Sequence Diagram

Interaction Fragments

Occurrence

Execution

State Invariant

Interaction Use

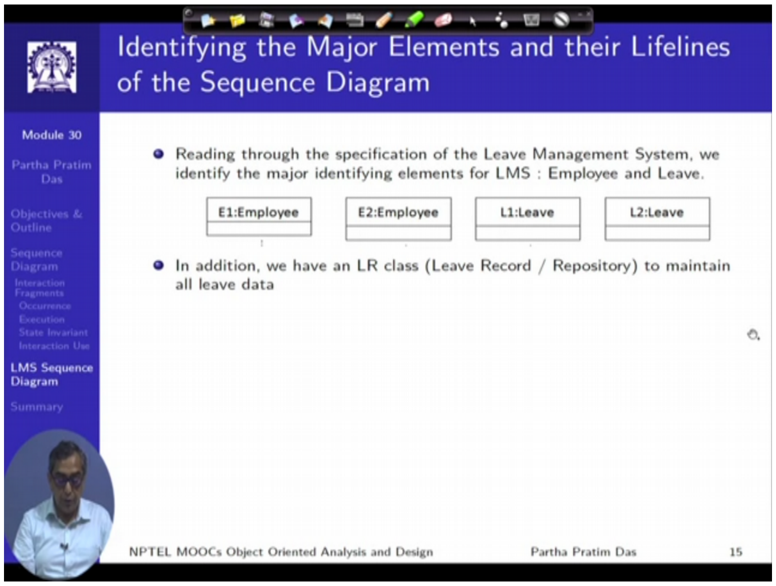
LMS Sequence Diagram

Summary

NPTEL MOOCs Object Oriented Analysis and Design Partha Pratim Das 14

I will leave this as an exercise for you to again try to follow. The Facebook authentication, you can see a fragmentation being done here I think there is no other no other interaction fragment used explicitly here of course you will have the occurrences executions I mean all ov all over the place. So, you can study them better.

(Refer Slide Time: 21:52)



Identifying the Major Elements and their Lifelines of the Sequence Diagram

Module 30

Partha Pratim Das

Objectives & Outline

Sequence Diagram

Interaction Fragments

Occurrence

Execution

State Invariant

Interaction Use

LMS Sequence Diagram

Summary

- Reading through the specification of the Leave Management System, we identify the major identifying elements for LMS : Employee and Leave.

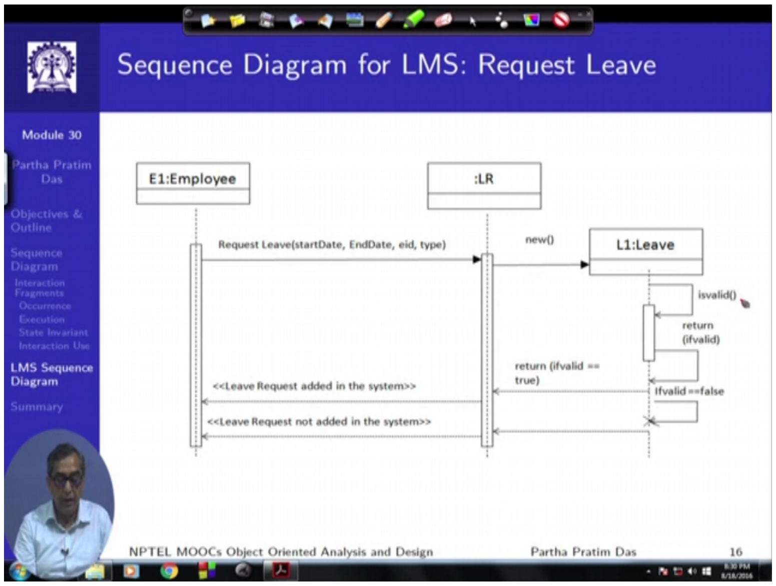
- In addition, we have an LR class (Leave Record / Repository) to maintain all leave data

NPTEL MOOCs Object Oriented Analysis and Design Partha Pratim Das 15

Now let us move on to using these in terms of our lms example in terms of the lms exercise. Now certainly we have been as we have been repeatedly coming through we have seen that there are 2 major named elements, identities in the lms the employees and the leave. So, while we want to build the sequence diagram for lms these will be the typical objects. So e1 colon colon employee which means that of the employee class, e1 is an object instance.

It gives another object instance. L1 is an object instance of leave and so on so forth. So, we will have these timeline lifelines on those. Eh in addition actually what we have not what is what was not explicit in terms of the specification of the leave management system but which is kind of a implementation class that we have to deduce is if there are so many leave fra requests ehh resign in the system with the different you know state information whether the leave has just been requested.

Or it is been approved or it is been regretted and so on then we need some kind of a store to keep this leave information. So, we say we will have a LR which we can say a leave record or a leave repository where all these collections of leave will exist and LR will also provide a major part in terms of the lifeline.
(Refer Slide Time: 23:30)



So, let us say if we just this is not a not a very detailed one as I said that I would like you to detail out each and every you know diagrams that we are doing for the lms system but this is just to show you how to go about things. So, here is the 3 major lifeline items, E1 anonymous LR and L1. So how will it start, it starts with the employee sending a request for leave, so for the request for leave there will have to be several parameters.

So, this message will have parameters like start date, end date, the employees' id, the type of leave that the employee wants, there will be more. But some basic parameters swill have to be there and whom would you send the request. Now the leave does not exist so that request have to go to the common repository store who is kind of providing the services of managing the leave. So, this is kind of a managing the leave that will assume. So once the LR gets this, then LR knows that this is a, so it has started this execution LR has started this execution.

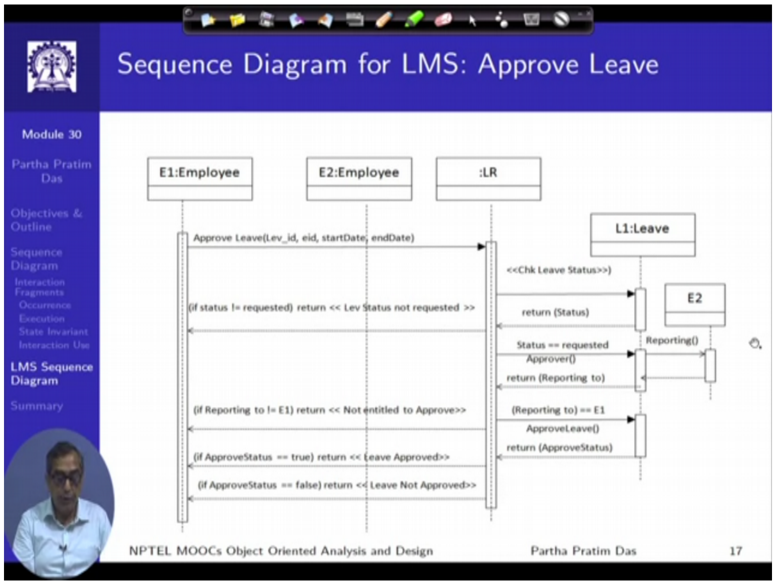
Employee was already in execution so we are not showing the earlier part of how the employee has looked the in and out application and all that. That will be a separate diagram again. But now LR has received the leave request so the LR has to create a leave record. So, you send a create message, here we have marked it as new and this leave record, this leave object gets created. Now we know that all kinds of leave are not valid in the system some are valid, some are invalid.

So, after creation this leave record sends some self-message saying is it valid. So, it invokes the message that it is valid so which is basically starts an execution to check if the leave is a valid leave or it is not a valid. So, it is expected to return some kind of a Boolean value that say if valid. So, when this return if got this is valid so it returns this so it is a possibly that if valid may be false with the possibility that if valid is true. Certainly, if in case if valid is true we infer that a leave object has been created with the given parameters and it is a valid leave.

So, the response goes back to the LR at this point saying that we have got a correct leave record now. So once LR gets that naturally LR will again send a response back a reply back that is leave request is added in the system and based on that again execution here will proceed. But it is possible that the leave itself is not valid so if valid is returned as false, if it is returned as false then what will need to do we will need to do 2 things, one is we will need to have a delete send a delete message to itself because this leave is not attainable one.

So, it has no reason for this to be maintained and all these returns if valid is false, return will come back to the LR again and when it comes back to the LR, LR now knows earlier it knew that it is true now it knows that the if valid is false that is the request to create the leave has not been successful. And that is why it got a false message. So, when it got a false message then it will again send a response back reply back to the employee saying that leave request not added in the system because it is not possible because the request was not a valid one.

So just a just a simplistic diagram as yet but it gives you the basic flavor of activities that has stated in the specification in terms of what is expected by an employee in terms of the request.
(Refer Slide Time: 27:55)


Sequence Diagram for LMS: Approve Leave

Module 30

Partha Pratim Das

Objectives & Outline

Sequence Diagram

Interaction Fragments

Occurrence

Execution

State Invariant

Interaction Use

LMS Sequence Diagram

Summary

E1:Employee
E2:Employee
:LR
L1:Leave
E2
Approve Leave(Lev_id, eid, startDate, endDate)
<<Chk Leave Status>>)
return (Status)
(if status != requested) return << Lev Status not requested >>
Status == requested
Approver()
Reporting()
return (Reporting to)
(if Reporting to != E1) return << Not entitled to Approve>>
(Reporting to) == E1
ApproveLeave()
return (ApproveStatus)
(if ApproveStatus == true) return << Leave Approved>>
(if ApproveStatus == false) return << Leave Not Approved>>

NPTEL MOOCs Object Oriented Analysis and Design   Partha Pratim Das   17

Now let us look at the next step if a leave has been requested at a point of time it will need to be approved. So, we perceive that there is an employee E1 who has to be a leader, a manager starts and approve leave, sends an approve leave message to the LR to the leave report which has all different parameters. So, there will be further details to that because we can always question that how does the employee know that such a leave exist.

So, will possibly assume that there is a list which is blindly shown to the leads and the manager and they can pick up any one of them with the leave id of the leave that is created, naturally associated by id, start date, end date all those leave properties. So once the approve leave is sent to is been received by LR at this point, the LR has to first check that the leave inquisition actually has an appropriate status so that it can be approved. So, if a request has come for a approval on leave which is cancelled.

That is not certainly acceptable. So, it sends a message to the leave, the leave object now to check the leave status. Executes and what is returned is the status of the leave. So, once it gets the status of the leave naturally there are 2 possibilities. One is the status is not requested leave, that is it is in some other state, it is cancelled or it is regretted or something. So, if the status is not requested then LR would immediately return a message ss send a response back to the employee saying that the leave is not in requested state and therefore there is no question of approving it.

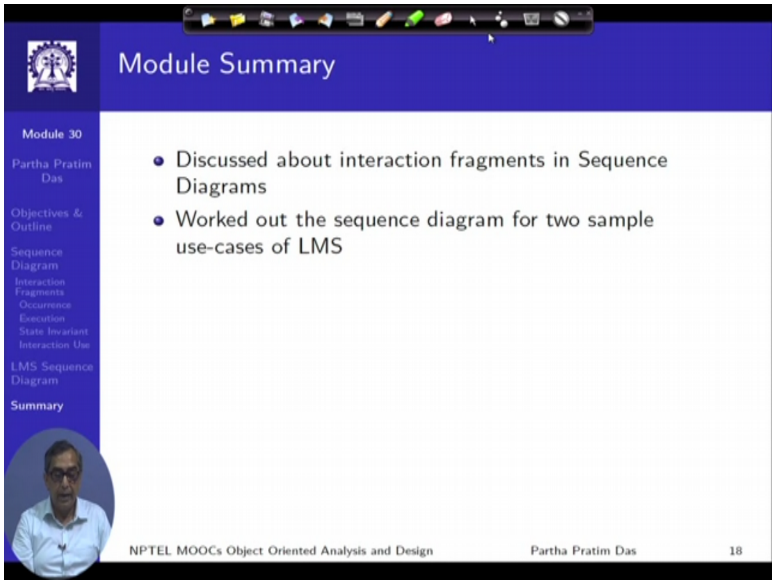
In the other case if the status is same as requested which means that the leave has been created but yet to be approved or regretted then it will send a approval message to the leave which says that we want to the LR as if is telling the leave that I want to initiate the approval process for this particular leave and I need adequate information further that. So, what is the requirement? We did a say in terms of the modeling constraints in the class diagram which we will use here.

In the sequence diagram is approval can be done only by those to the only by that particular employee to which the leave applicant reports. So certainly, this execution of approval process has to invoke another message and that message is goes to E2 which will, so E2 is the employee who had requested for the leave. So, leave will send the message there and get back response from the employee as to who is the reporting person, reporting manager for E2.

Now once that has been received here, the reporting to, the information of the reporting person then the LR will have to check whether this is same as employee E1. Because employee E1 has initiated the approval so if it is E2's leave that E1 is planning to approve then E1 needs to be the reporting manager for E2. So, if the reporting manager is same as E1 then certainly the LR proceeds with the approved leave process on the leave L1 and that is possibly approved or it may not be approved because of other reasons, documentation reasons, duration reasons and so on and then send that status will return.

So, in case the reporting to is not E1 then certainly this part cannot proceed anymore so the LR will send a message back that reporting is not equal to E1 so not in entitled to approve the leave. Whereas if it is successful then the approval request will go and at the end of this execution from the leave object will get the approval status then depending on whether the approval status is true that is the leave can be approved then you say leave approved or if the approval status is false then you said leave not approved.

So, based on these conditions one of these 2 response messages will be sent back to the employee. So, this is just showing up the glimpse of how we can visualize the approved leave dynamics behavior so I would request that like this you know work on the other lea use cases and build up the sequence diagram for the cancelled leave, regret leave, reject leave, revoked leave and so on and that way try to complete the sequence diagram for the LMS.
(Refer Slide Time: 33:34)



Module Summary

Module 30

Partha Pratim Das

Objectives & Outline

Sequence Diagram

Interaction Fragments

Occurrence

Execution

State Invariant

Interaction Use

LMS Sequence Diagram

Summary

- Discussed about interaction fragments in Sequence Diagrams
- Worked out the sequence diagram for two sample use-cases of LMS

NPTEL MOOCs Object Oriented Analysis and Design Partha Pratim Das 18

So, in summary here in this module we have completed a discussion on sequence diagram having talked of having interaction fragments and then we have taken a quick look into the sequence diagram of the LMS and for 2 use cases we have outlined how the sequence diagram will look like.