

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

Lecture – 43
Communication Diagram

Welcome to module 31 of object-oriented analysis and design. We have been discussing about uml diagrams, we have covered 3 major diagrams the use case, the class diagram and the sequence diagram. In the current module, we will talk about communication diagram which is also an interaction diagram like the sequence diagram.

(Refer Slide Time: 00:47)

The screenshot shows a presentation slide with a blue header and sidebar. The header is titled "Module Objectives". The sidebar on the left lists the contents of Module 31, including "Objectives & Outline", "Communication Diagrams", "Frames", "Lifelines", "Messages", "Examples", and "Summary". The main content area of the slide lists the objective: "Understanding Communication (Collaboration) Diagrams". The footer of the slide contains the text "NPTEL MOOCs Object Oriented Design and Analysis", "Partha Pratim Das", and the page number "2".

And it has a lot of relationship with the sequence diagram as well.

(Refer Slide Time: 00:52)

The slide is titled "Module Outline" and is part of "Module 31" by Partha Pratim Das. The left sidebar lists the following items: "Objectives & Outline", "Communication Diagrams", "Frames", "Lifelines", "Messages", "Examples", and "Summary". The main content area lists the topics for the module:

- What are Communication (Collaboration) Diagrams?
 - Frame
 - Lifeline
 - Messages
 - Examples

A small circular video inset of the speaker is in the bottom left. The footer includes "NPTEL MOOCs Object Oriented Design and Analysis", "Partha Pratim Das", and the slide number "3".

So, will talk about what is the communication diagram and what are the different components of that diagram

(Refer Slide Time: 00:59)

The slide is titled "Client-Server Computing Model: RECAP (Modules 05, 11)" and is part of "Module 31" by Partha Pratim Das. The left sidebar lists the following items: "Objectives & Outline", "Communication Diagrams", "Frames", "Lifelines", "Messages", "Examples", and "Summary". The main content area lists the key points of the client-server model:

- No object exists in isolation
- Objects are acted on and themselves act on other objects
- Leads to the **Client-Server Model** of computing where
 - Behavior is
 - Services provided by an object
 - Services are requested by
 - Sending Messages, Invoking Operations
 - In Client-Server View
 - Clients request for Services
 - Servers provide Services
 - Contract between client and server ensures correctness

A small circular video inset of the speaker is in the bottom left. The footer includes "NPTEL MOOCs Object Oriented Design and Analysis", "Partha Pratim Das", and the slide number "4".

I would just like to remind you that for understanding this module well, you should be well-versed with the cli client-server computing model which we have discussed earlier.

(Refer Slide Time: 01:10)

Communication (Collaboration) Diagrams in SDLC phases:
RECAP (Module 22)

Module 31
Partha Pratim Das

Objectives & Outline
Communication Diagrams
Frames
Lifelines
Messages
Examples
Summary

```

graph LR
    AM[Analysis Model] -- 1 --> SM[Structure Model]
    AM -- 1 --> BM[Behavior Model]
    SM -- 1 --> CD["«UML»  
Class Diagram"]
    BM -- 1 --> SD["«UML»  
Sequence Diagram"]
    BM -- 1 --> CD2["«UML»  
Collaboration Diagram"]
    BM -- 1 --> SCD["«UML»  
Statechart Diagram"]
    BM -- 1 --> AD["«UML»  
Activity Diagram"]
  
```

- In the **Analysis Phase** the problem domain is analyzed and refined from the **Requirements Phase**
- The behavior model of the system is hence understood in this phase
- Communication (Collaboration) diagrams is a result of the Analysis Phase

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 5

And there are communication diagrams and they just note that we are often referring to it as collaboration diagram also because as history goes, there in the uml version of 1.x somewhere in the middle, this diagram was introduced as a collaboration diagram. Later on, when modifications were made and we got into uml 2.x currently that same diagram is been renamed as a communication diagram.

So, in the community both terms are interchangeably used. So, this diagram, this collaboration or communication diagram offers as a part of a behavioral diagram, the interaction diagram in the analysis stage that is where it gets generated and then it gets refined in the design phase as well which we had seen this earlier.

(Refer Slide Time: 02:07)

Communication (Collaboration) Diagrams in SDLC phases:
RECAP (Module 22)

Module 31
Partha Pratim Das

Objectives & Outline
Communication Diagrams
Frames
Lifelines
Messages
Summary

```

graph LR
    DM[Design Model] -- 1 --> SM[Structural Model]
    DM -- 1 --> BM[Behavioral Model]
    SM -- 1 --> CD["«UML»  
Component Diagram"]
    SM -- 1 --> DD["«UML»  
Deployment Diagram"]
    BM -- 1 --> AD[Architectural Description]
  
```

- Communication (Collaboration) diagram is included in the Behavioral Model
- It is further refined in the **Design Phase**

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 6

So, in the design phase it is actually bundle in terms of this particular behavioral diagram.
(Refer Slide Time: 02:17)

The screenshot shows a presentation slide titled "What are Communication Diagrams?". The slide is part of "Module 31" by Partha Pratim Das. The left sidebar lists "Objectives & Outline" with "Communication Diagrams" as the current topic, followed by "Frames", "Lifelines", "Messages", "Examples", and "Summary". The main content area lists four bullet points: 1. Communication Diagram (called Collaboration Diagram in UML 1.x) shows interactions between objects and / or parts (represented as lifelines) using sequenced messages in a free-form arrangement. 2. Communication Diagram is a UML Behavior Diagram. 3. Communication Diagram depicts the inter-object behavior of a system, ordered by space. 4. The major components of a Communication Diagram are: Frames, Lifeline, and Messages. A handwritten diagram in red ink shows a sequence of objects: $O_1 \xrightarrow{m} O_2 \xrightarrow{m'} O_3 \xrightarrow{m''} O_4$. The bottom of the slide includes the NPTEL MOOCs logo, the text "NPTEL MOOCs Object Oriented Design and Analysis", the name "Partha Pratim Das", and the number "7".

So now what are communication diagrams, the communication diagrams is basically interaction, as the name suggests it saying communication diagram. So how does objects communicate between themselves. So, looking at the client server model basic computation is message is through messages, one object has to send message to another and in the recipient object has to react to the. So, in the communication diagram, we try to check this interaction between the different objects which are known as lifelines.

The messages are sequenced and we will see this more what the sequencing mean. do not get confused with when we said the sequencing diagram. The sequence diagram talked about the sequencing which is strictly temporary in nature. So, we were always concerned in the sequence diagram in the terms of the looking at the messages in the order in which they happen in time. In communication diagram, in contrast we will try to look at what is saying as more in the order of space.

That is we would be more interested if object o1 has sent a message m to object o2, then eventually when will object o2 send a message to object m3 and when object m3 will send a message to object o3 will send a message to object o4 and so on and it is not necessary that they will be in consecutive time bound but what we are trying to see is basically what is covered by space that is how objects actually receive and send messages, how do they communicate and that is the basic purpose of the communication diagram.

(Refer Slide Time: 04:18)

Frames

Module 31
Partha Pratim Das

- Communication diagrams could be shown within a rectangular frame with the name in a compartment in the upper left corner

Interaction Frame for Communication Diagram
BuyItem

Sd Frame (short frame) for Communication Diagram
Diagram BuyItem

Source: UML 2.5 Diagrams Overview: <http://www.uml-diagrams.org/uml-25-diagrams.html> (20-Aug-16)

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 8

Typically, this diagram is drawn in terms within frames which are rectangles and that the corner, we have a there are 2 styles of naming, either you say interaction and then say the name of the communication diagram or you say SD and then you say the name of the diagram. SD is little bit of a misnomer it derives from the sequence diagram term but this actually means a communication diagram. So, this is how you will know that this is the communication diagram.

(Refer Slide Time: 04:45)

Lifelines

Module 31
Partha Pratim Das

- Lifeline is a specialization of named element which represents an individual participant in the interaction
- A Lifeline is shown as a rectangle (corresponding to the head in sequence diagrams)
- Lifeline in sequence diagrams does have tail representing the line of life whereas lifeline in communication diagram has no line, just head
- The information identifying a lifeline is depicted as **ObjectName[selector]:ClassName**

Anonymous lifeline of class User

Lifeline "data" of class Stock

Lifeline "x" of class X is selected with selector [k]

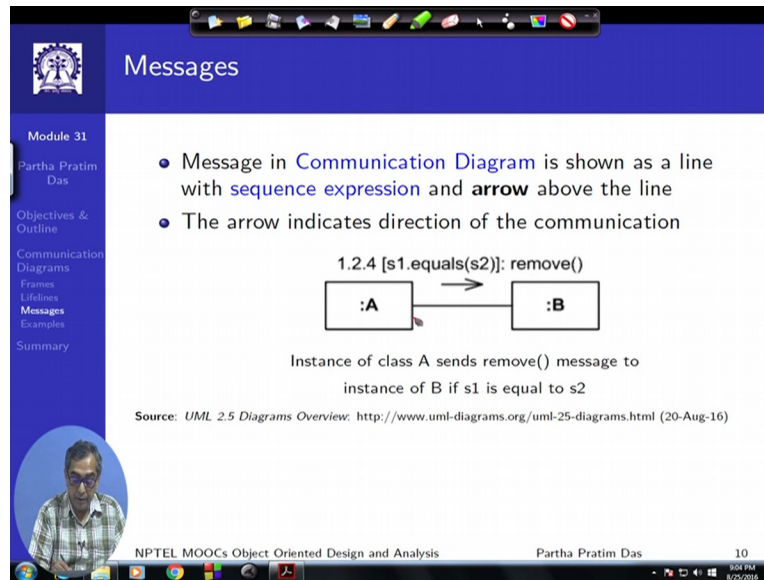
Source: UML 2.5 Diagrams Overview: <http://www.uml-diagrams.org/uml-25-diagrams.html> (20-Aug-16)

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 9

Then you have lifelines as as we had in the sequence diagram in you recall this a lifeline is an instance of an object and we had this that it is a box within which the name of the object is given and then we have the vertical actual line over which the object has a life. The lifeline in communication object just has a rectangle or the it is called head of it, it does not have the tail. Its tail part is not there, it just has the head so that is called the head in terms of the communication diagram.

In sequence diagram, you have the head as well as the tail, so these are typical heads that you have, we had seen this similar head in sequence diagram, this is an anonymous instance of the user class, this is an instance named data of the stock class and this is a collection data where am we are talking about the kth lifeline of the x collection. So, these are the basic terminology.

(Refer Slide Time: 05:42)



Then we have messages. Messages are shown in terms of a connection line and you show a arrow designating who is sending message to whom. So, this is an instance of class a, this is instance of class b and this is the message that being sent. You will see that we will next go into what does this number mean, it is called the sequence expression and this the condition which says s1, if s1 is equal to s2 then the remove message will be sent.

So that is the basic structure of the message. Now the how the messages are sent or in what condition what message is sent is guided by the whole of the sequence expression which you see here.

(Refer Slide Time: 06:30)

Sequence Expression

Module 31

Partha Pratim Das

Objectives & Outline

Communication Diagrams

Frames

Lifelines

Messages

Examples

Summary

- The sequence expression is a dot separated list of sequence terms followed by a colon (":") and message name after that:
`sequence-expression ::= sequence-term '.' . . . ':' message-name`
- Example: **3b.2.2:m5** : Sequence expression **3b.2.2** and message name **m5**
- Each Sequence term
`sequence-term ::= [ integer [ name ] ] [ recurrence ]`
- The **integer** represents the **sequential order** of the message within the next higher level of procedural calling

Source: UML 2.5 Diagrams Overview: <http://www.uml-diagrams.org/uml-25-diagrams.html> (20-Aug-16)

NPTEL MOOCs Object Oriented Design and Analysis

Partha Pratim Das

11

So, you will now next go into motion of sequence expression. So, the sequence expression, this is the this is the formal definition of the sequence expression or will just look at this. So, in the sequence expression you have 2 parts one is a actual expression and the other is a name of the message, they are separated by a colon. So, this is this tell me what is the sequencing of the message and in what way and this tells me what is the message.

So, these are 2 parts, sequencing and the message separated by colon. So, this integer basically these integers basically tell me the sequential order of the messages.

(Refer Slide Time: 07:11)

Sequence Expression: Sequential Order

Module 31

Partha Pratim Das

Objectives & Outline

Communication Diagrams

Frames

Lifelines

Messages

Examples

Summary

- Example:**
 - message with sequence 2 follows message with sequence 1
 - 2.1 follows 2
 - 5.3 follows 5.2 within activation 5
 - 1.2.4 follows message 1.2.3 within activation 1.2.

Instance of A sends draw() message to instance of B, and after that B sends paint() to C

Source: UML 2.5 Diagrams Overview: <http://www.uml-diagrams.org/uml-25-diagrams.html> (20-Aug-16)

NPTEL MOOCs Object Oriented Design and Analysis

Partha Pratim Das

12

So, let us look into some examples. In this sequential order, it is strictly integer order. So, a sequence 2 follows a sequence 1 and then in effect of 2 if you are doing a new message next, then you mark it as

2.1 which will follow the action 2. So, 5.3 will follow 5.2 in the message when the activation is for 5. 1.2.4 will follow 1.2.3 then the activation is for 1.2. so, it keeps on going like that. So, in c the messages in space but through the sequence numbers you can relate as to what is their ordering in the time domain.

So, the purpose like like in sequence diagram if you if you recall we had the lifelines going like this and going like this and and basically, we as go on messages on this on a on an order. So, this temporary order was a important thing, then of course you could see the special one because the different lifelines, I have one lifeline here, I have another lifeline here and I can say that this sends a synchronous message to this lifeline.

So, we could see who is sending what message to which other object but the interestingly the sequence diagram is looking into this time order. And interestingly the communication diagram is to look into the special order that who is sending to whom and the sequencing information is encoded in the sequence expression. So here we wa if we read this sequence diagram it says that the instance of a is sending a message draw message to instance of B which is, so sequence order is 1.2

And on receiving that, B sends a message paint to an instance of C and there its sequence expression turns out to be 1.3. so that is how by reading this sequence diagram you can figure out in which order they are happening

(Refer Slide Time: 09:21)

The slide is titled "Sequence Expression: Concurrent Thread". It features a blue header with the title and a sidebar on the left with navigation links: "Module 31", "Partha Pratim Das", "Objectives & Outline", "Communication Diagrams", "Frames", "Lifelines", "Messages", "Examples", and "Summary". The main content area has a white background with a blue border. It contains a bulleted list: "The **name** represents a **concurrent thread** of control" and "Example: messages 2.3a and 2.3b are concurrent within activation 2.3". Below the list is a sequence diagram showing a lifeline :A sending two concurrent messages, 2.3a:draw() to :B and 2.3b:draw() to :C. A note below the diagram states: "Instance of A sends draw() messages concurrently to instance of B and to instance of C". At the bottom, the source is cited as "UML 2.5 Diagrams Overview: <http://www.uml-diagrams.org/uml-25-diagrams.html> (20-Aug-16)" and the slide is identified as "NPTEL MOOCs Object Oriented Design and Analysis" by "Partha Pratim Das" with the number "13".

Sequence Expression: Concurrent Thread

- The **name** represents a **concurrent thread** of control
- **Example:**
 - messages 2.3a and 2.3b are concurrent within activation 2.3
 - 1.1 follows 1a and 1b
 - 3a.2.1 and 3b.2.1 follow 3.2

Instance of A sends draw() messages concurrently to instance of B and to instance of C

Source: UML 2.5 Diagrams Overview: <http://www.uml-diagrams.org/uml-25-diagrams.html> (20-Aug-16)

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 13

But the communication diagram primarily has to specify what is the special order, what is the object

interaction order. at times you are you can use a name along with this number, it is a typically a symbol alphabet. If you use that then it gives concurrent messages. So here what is meant is there is an instance of a class A, it is sending a message draw to an instance of class B, it is sending a message draw to instance of class C and these are happening at a sequence order of 2.3.

So, in this, that is in the activation of 2, first 2.1 has happened, then 2.2 has happened, then in this the 2.3 A and 2.3 B, these are happening in concurrent one, that is between these 2, between these 2, there is no sequential order. They will happen at a same time, so this is what you specify when you write this named part along with the sequence expression. So, if I have 1a, 1b that means 1 who is taking place concurrently in terms of 1a and 1b and when both of these are over then we will have the sequence expression 1.1.

So that is the that is the basic so you you first resolve without the name and when once you have resolved without the name, then if the integer expression is same, the names may that they are all concurrent messages that will happen.

(Refer Slide Time: 11:03)

Sequence Expression: Guard

- A **guard** specifies condition for the message to be sent (executed) at the given nesting depth
- **Example:**
 - **2.3b [x>y]: draw():** message draw() will be executed if x is greater than y
 - **1.1.1 [s1.equals(s2)]: remove() –** message remove() will be executed if s1 equals s2

Diagram illustrating a message from instance :A to instance :C with the guard condition **2.3b [x>y]: draw()**.

Instance of class A will send message draw() to the instance of C, if $x > y$

Source: UML 2.5 Diagrams Overview: <http://www.uml-diagrams.org/uml-25-diagrams.html> (20-Aug-16)

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 14

Messages can have guards, guards are like Boolean conditions, so here we say that this is a am sorry, we say that this is a guard, so guard is x greater than y. so it says that at 2.3 sequence at the concurrency b so there is some something else which is concurrency a. if x is greater than y then send that draw message. So, it will be executed if x is greater than y. similarly here is another sequence expression is 1.1, at that level if s1 equal s2 then you send the remove message. So, this is these are the guard conditions that you can specify in terms of sending the messages.

(Refer Slide Time: 11:50)

Sequence Expression: Recurrence and Iteration

Module 31
Partha Pratim Das
Objectives & Outline
Communication Diagrams
Frames
Lifelines
Messages
Examples
Summary

- The **recurrence** defines conditional or iterative execution of zero or more messages that are executed depending on the specified condition
 $\text{recurrence} ::= \text{branch} \mid \text{loop}, \text{branch} ::= '[' \text{guard} ']'$
- An **iteration** specifies a sequence of messages at the given nesting depth
- Notation:**
 - * : Messages Executed Sequentially
 - *|| : Messages Executed Concurrently

Source: UML 2.5 Diagrams Overview: <http://www.uml-diagrams.org/uml-25-diagrams.html> (20-Aug-16)

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 15

Certainly, the messages could be sent in a recursive or a iterative manner. You can keep on sending number of messages to number of objects from the at the same level and when you send number of messages to number of objects you can have 2 protocols again. That you could send those messages in sequential manner, one after the other or you can send the messages in concurrent manner that is all together.

(Refer Slide Time: 12:23)

Sequence Expression: Recurrence and Iteration

Module 31
Partha Pratim Das
Objectives & Outline
Communication Diagrams
Frames
Lifelines
Messages
Examples
Summary

Example:

- 4.2c * $[i=1..12]$: search(t[i]) – search() will be executed 12 times, one after another
- 4.2c *|| $[i=1..12]$: search(t[i]) – 12 search() messages will be sent concurrently
- 2.2 *: notify() – message notify() will be repeated some unspecified number of times

Source: UML 2.5 Diagrams Overview: <http://www.uml-diagrams.org/uml-25-diagrams.html> (20-Aug-16)

Handwritten notes: *

Handwritten notes: search (t[1]), search (t[2]), search (t[12])

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 16

Example let us say am saying 4.2c so 4.2c means it is at a it is the second activation, second message in the activation 4. It is the c concurrent thread of the execution and in this am saying star I 1to 12 send the message, search it which means that the fact if I just write this means sequential. So, it says that from 1 to 12, you keep on sending search messages. So, you first search, search t1, then you send

search t2 and so on. You execute them one after the another.

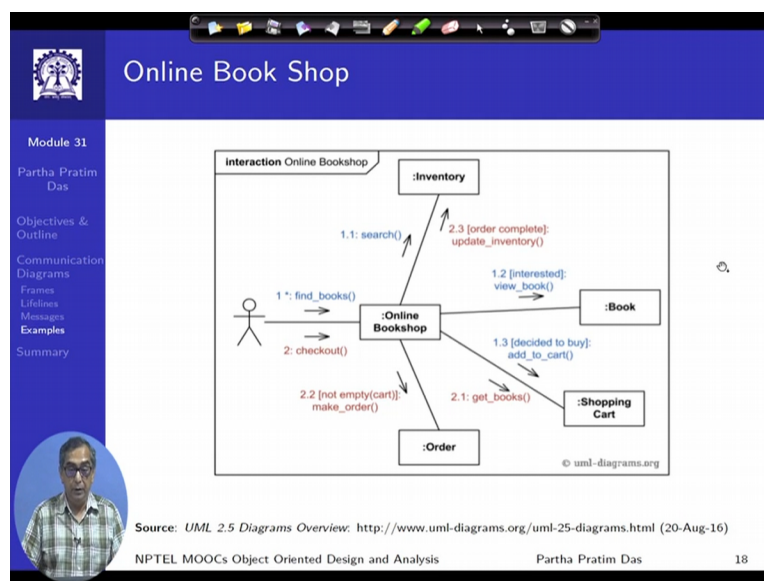
But when this whole thing if I introduce to parallel bars that means it is a concurrent message which means that all these 12, search t1 to search t12 all these concurrent messages will be sent together, they will be sent at the same time. In this case, it will be sent 12 times one after the other, in this case they will be sent at the same instance. So, this is how these are 2 different kinds of iterations, sequential or concurrent iterations that you can specify.

You may not specify how many times a message will be iterated, so it could be like just the star, so it says that at 2.2 level the notify message will be sent to, you do not know how many, star means it could be 0, it could be any number. It will be repeated arbitrary unknown number, unspecified number of times. It may be governed by some other fact.

(Refer Slide Time: 14:20)

So, this is these are examples of the iteration, for example you have already explained this is a going from these are n different requests being sent one after the another, this n different request being sent concurrently at this runtime.

(Refer Slide Time: 14:43)



So, let us look at an example to understand all that we have. So just to show you, I will, there is an annotated diagram also which has bring up for your practice later on. So, these are the different lifelines we have, just the heads, the online bookstore, the inventory, the book, shopping cart, the order all the kind of different objects that we need if you want to buy some book from an online bookshop.

If you look at these you can certainly see that these are the different messages that are that are involved here, just for convenience of and better understanding, I made the level 1 messages as blue and level 2 messages as red, otherwise they do not have any specific meaning in terms of uml. So, from these numbers you can figure out certainly the first message is this, where the actor this you know is the actor, sends message to the online bookshop and what is the message, find books.

What is the nature of this message, you can see from this star that it is a repeated message, it is an iterative message. What kind of iteration, this is a sequential iteration. So, what does this message being sent from the actor to the online book store instance mean to start with, it means certainly as you can understand if the actor is trying to find books repeatedly, one after the other. Now what what what happens once once the actor has sent this to the online store.

Online store in turn has to look up the inventory that thee books are there. Because the online store is store will have to see whether the book is available, so based on one of these messages in this sequence, it will fa in fact for each one of these messages one by one in the sequence, it will be 1.1 that is a next message to send which is a search message which will go to inventory and then inventory will find it not find it, whatever. Based on that then 1.2 is so now you try to perceive what is happening.

The the actor is sitting in front of the online book store window, so to find books so there is some specify some keywords, some author names something, to find books which is taken by the online book store, then a message is sent to the inventory for the search and you have got a set of lists possibly and then the question is whether you select a book depends on if it is interest. If there are 10 books come up, you do not look at all books. So, you do a view book of a particular book object provided you are interested.

So, you model that by saying that the online bookshop sends a message to thee instance of the book which is guarded by the fact that if the person is interested and the message is view book. And look at this was 1, this is 1.1, this is 1.2, so logically this is the next this that will happen. If you look into this in the sequence diagram then you will see this 1 first, then you will see 1.1 and the and the different life lines you will see see them in this order. But here you are seeing them in the special order of who is talking to whom for the task.

Now certainly once this has been done, the interested book has been booked up then in some cases, the person would be interested to buy, would be would decide to buy. So, 1.3 is a next message which is guarded by decide to buy. If the book the persons was interested in, the person is deciding to buy that, if that condition gets satisfied then the book gets added to the cart. So that you say add to cart. So, this becomes in this sequence this becomes a next message.

So here you can clearly see that 1 is a overall activation of finding a book and that has been achieved through 3 sub activations of 3 messages of search from online bookshop inventory, view book from online book shop to book and finally add to cart from online book shop to the shopping cart. So, this kind of complete one activity and then certainly then what will happen again this is this is got a star, this is a repeated iterative process, so this process will keep on happening every time you send this out,

Send the find book out, all all messages may happen not all may be all messages will not, this this may be false, this may be false, so in that case the message will not get send. But in this order, this will keep on happening, keep on happening, resulting in this shopping cart I think filled by the books that the persons has decided to buy. So, what is the next activity that will happen? What is the next thing that is going to happen is mark a sequence 2 which is a checkout process.

So, you said. Now the actor sends a checkout message to the online bookshop. The online bookshop knows that now the checkout has to happen so why will that checkout will happen for, naturally from the shopping cart. So, at this level the next sequence level is 2.1 and that is to get thee books from the cart. Certainly, there is no guard condition because if you have been asked to checkout whatever there in the cart will be blocked for the consideration of the person.

Of course, you can refine and make some you know some hygiene conditions as guards for example, get books will apply provided you have the some books in the shopping cart. Now once so that you will you will see subsect widely so once that has happened then you have got the books that there in the cart so this is the next message in the cart is empty our not. Certainly, if the cart is empty and there is no order to make so this message cannot be sent.

But if the cart is not empty definitely if the cart has one book then will proceed to make the order which is a message that the online book store, bookshop will send to the instance of the order order activities. So once the order is ready then so with they will they will be several several other examples

and the you reach a state where the order is complete, this condition becomes true, then you have the next sequence in this in the activation 2 after 2.2 is 2.3 where you say if the order is complete that is the guarded one, then the inventory because they will have to be shipped. So, this is just to show and explain to you as to how the em different actions in the online bookshop can be represented in terms of messages and and how these messages are sequenced for the purpose of representation.

Online Book Shop - Annotated

Module 31

Partha Pratim Das

Objectives & Outline

Communication Diagrams

Frames

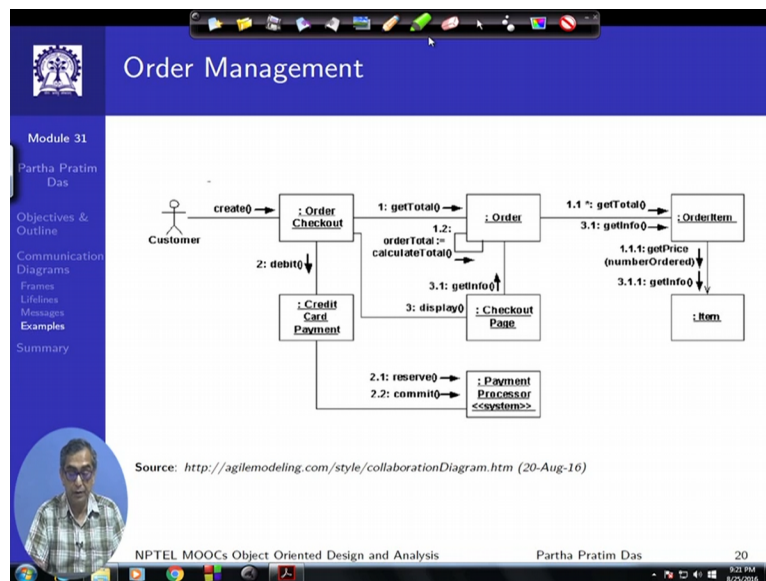
Lifelines

Messages

Examples

Summary

(Refer Slide Time: 22:59)



There is another example here for order management which have primarily included the for your practice. So, your customer who places an order so that is for order checkout part primarily and so this is this does not have a sequence expression which means this is what starts the whole thing then these are first action which is getting totaled. So, if you have to check out the order means ehe you have to actually start considering how much you have to pay.

So, if the order checkout sends a get total message to the order instance. You can see that in compared to the earlier one this is little differently for example here for every message there is a function operator notation which is used. So, these are all optional stuff and after this has get total has happened, naturally the order object need to know what are the items that exist in the order. So, it sends after this 1, it sends the next sequence is 1.1 which is get total sent to the order item.

This other item need to tell what is the total because it depends on the item. Now the item cannot say that it needs to the order item cannot say that whether it needs to actually go to the item in the inventory so it sends 1.1.1 which is getting the price depending on how many have been ordered. So that is the parameter that you send that these many have been ordered and these are next message. So once that has happened so this then within this within this

Then you go to the next stage in the processing which is after 1.1 next stage is 1.2 that is you get the total of the order because you have now got the price for each and every order item so your 1.2 is the total of the order which the order sends to itself. So here you can see that these are self-message kind of, we have seen self-message earlier in sequence diagram also so this is the self-message which order

sends itself to find out that what is the total of this order.

So now after having done this, you have the total of the order, so next stage would be to so you have done with the messages starting with 1, so next activation is to which is that money the order total money has to be paid. So, you send a debit message to the credit card payment that within that activation you have to reserve block that with the payment processor which is the payment gateway or something.

After it has been blocked and your card details, identity has been authenticated, then it has to commit that is actually debited from your available balance in the card if the commit message happening and that will complete this part which is debiting the order total. So, the action 1 was activation 1 was done, now activation 2 has been done. Now what you expect, now you expect that all these information will come on the display and the detail on the items that you have actually purchased now through this order will be known to you.

So, the third activation is the display which the order checkout again send to the checkout page. Checkout page needs to know what the order is, sends out 3.1 to get that info to order which in turn goes to the order item which in turn goes to the particular item and through this 3, 3.1 and 3.1.1 you get the list of items which are displayed on the checkout page. So, this is just another kind of so all these sequencing you can see that here dominantly you are showing

What are these different lifelines which are interacting between these with this messages to achieve the different purpose that they need to serve and at the same time you are following certain sequence expression which are show the original temporal ordering of their behavior. So, the same thing drawn in the sequence diagram would look very different because these are based on these numbers thee messages will become order on the tail of the lifeline and these objects will not be placed according to their interaction but they will be placed just arbitrarily as vertical lifelines one after the other.

(Refer Slide Time: 28:08)

Module Summary

Module 31
Partha Pratim Das
Objectives & Outline
Communication Diagrams
Frames
Lifelines
Messages
Examples
Summary

- Communication Diagrams are introduced
- Various components of Communication Diagrams like Frames, Lifeline, Messages are discussed
- Examples are illustrated

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 21

So, this is the basic fundamental to the communication diagram. So, in this module we have introduced the communication diagram, we have ref we have made references to how closely it relates to the sequence diagram and what is the interoperability, the sequence diagram is the temporal behavioral diagram for interaction and communication diagram is a special behavioral diagram of interaction and both of them are closely related but they define 2 different aspects of the dynamic behavior,

We have discussed also couple of examples and I would em encourage you to we have covered it in the presentation here but I would encourage you to try to build the communication diagram for our leave management system exercise.