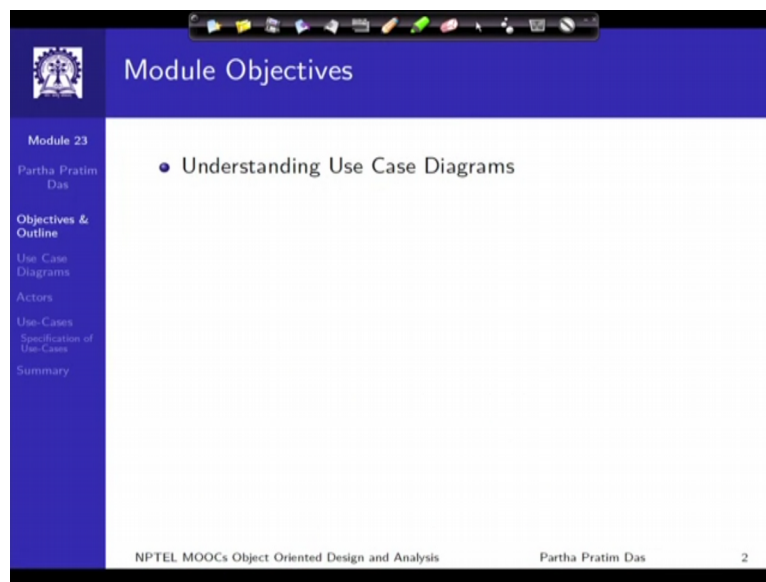


**Object-Oriented Analysis and Design**  
**Prof. Partha Pratim Das**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology-Kharagpur**

**Lecture – 35**  
**Use-Case Diagrams Part I**

Welcome to module 23 of object oriented analysis and design. We have already in the last 2 modules, taken a look at the basic premises of uml, the unified modeling language. We have discussed very briefly though the stages of software development lifecycle and related both of them together.

(Refer Slide Time: 00:54)



Now from this module onwards and this journey will continue for several modules to come. We will start discussing and explaining about different specific uml diagrams. So, in this module we start the discussion on use case diagram and we will continue that in the next module as well.

(Refer Slide Time: 01:20)

The slide is titled "Module Outline" and is part of "Module 23" by Partha Pratim Das. It lists the following topics in a bulleted format:

- Use Case Diagrams
- Actors
- Use Cases
  - Specification of Use-Cases

A sidebar on the left lists the module's structure: Module 23, Partha Pratim Das, Objectives & Outline, Use Case Diagrams, Actors, Use Cases, Specification of Use-Cases, and Summary. A small video inset of the presenter is in the bottom right corner.

So, in terms of our discussion we will discuss use case diagrams and specifically actors and use cases, the outline will be available on the left always.

(Refer Slide Time: 01:29)

The slide is titled "What are Use case Diagrams" and is part of "Module 23" by Partha Pratim Das. It lists the following points:

- The integration of business knowledge with the development specification is a requirement
- The development organization knows the specifications for developing a module, but does not know who will interact with these modules and for what purpose
- Use Case Diagrams depicts the human interaction with the system to give the context of who uses, which part of the system and for what purpose

Below the text, there is a hand-drawn diagram in red ink showing a stick figure (actor) connected by a double-headed arrow to a circle labeled "System". The sidebar on the left is identical to the previous slide, but "Use Case Diagrams" is highlighted. A small video inset of the presenter is in the bottom right corner.

So just to remind what are the use cases? Is a integration of business knowledge with the development specification is a requirement? That is that is the basic starting point. So that is how we got the 2 pages on the leave management system where the organization decided to have a system of lms and they put it down, integrated the business knowledge which is how they how the employees report, how the employees apply for leave, what different kinds of leave they have and so on and created this requirement.

Now we still do not know who will interact with these modules and for what purpose. What user will

do with this is as still not clear, certainly these are not so well specified. In fact, the lms specification is actually not that bad because if you back to the specification you will find the use of the word use cases. But usually you will not get that. So, the use case diagram depicts a human interaction with the system that is this is my system that am trying to build.

This is my system and finally why am in building the system because there is some human who wants to interact with the system, give input, take output and so on. And this is this human is not a not a single individual, this human could be a various different group of users who would use and different parts of the system for different purposes. Certainly, in any reasonable system, no 2 users would be using the system exactly in the same way or may be at least there would be variety of groups which will use the system in multiple different ways and the different ways in which the system can be used is the main premise of the use case diagram.

(Refer Slide Time: 03:40)

**What are Use case Diagrams**

Module 23  
Partha Pratim Das  
Objectives & Outline  
**Use Case Diagrams**  
Actors  
Use-Cases  
Specification of Use-Cases  
Summary

- Use case diagrams are behavior diagrams used to describe a set of actions (use cases) that systems perform in collaboration with one or more external users of the system (actors)
- Use case diagrams are mainly used to gather requirements of a system and identify the internal and external factors influencing the system.

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

So, it is a behavioral diagram as we already said and it is used to define a set of actions, use case would mean it is a particular case of use or cases of action that we are interested to look at and who takes an action, if an action who will take that action, an action will be taken by the actor. So, the use case diagrams are mainly used to gather requirements of the system and identify internal and external factors. Now before wasting more time let's get into the actual details.

(Refer Slide Time: 04:21)

**Components of Use case Diagrams**

The use case diagram is composed of

- Actors
- Use cases and their specifications
- Relationships between Use cases

Module 23  
Partha Pratim Das  
Objectives & Outline  
**Use Case Diagrams**  
Actors  
Use-Cases  
Specification of Use-Cases  
Summary

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

A use case diagram is composed of 3 major components, the actors, the use cases and the relationships. In simple terms, the actor is a user, in a simplistic term the actor is a user who wants to do something with the systems, use case is an action is an activity, or is a way that the user would like to use the system and finally relationship is no action that the user would want to execute would be all realizable alone by itself, different activities have different dependents.

If I want to take a leave then it depends on what is my availability of leave, what is the validity constraints on the leave, what is the opinion of my reporting manager and so on. So, a relationship is most important to understand what how the actions will interact.

(Refer Slide Time: 05:25)

**Actor**

- Actors are entities that interface with the system
- They can be people or other systems
- Actors are depicted as stylized stick figures
- This stick figures are used as stereotypes to depict many models at the same time like process, thread, meta-class, power-type etc labeled with guillemets <<>>
- The roles the actors play in the system is important, not their real world identity

Module 23  
Partha Pratim Das  
Objectives & Outline  
Use Case Diagrams  
**Actors**  
Use-Cases  
Specification of Use-Cases  
Summary

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

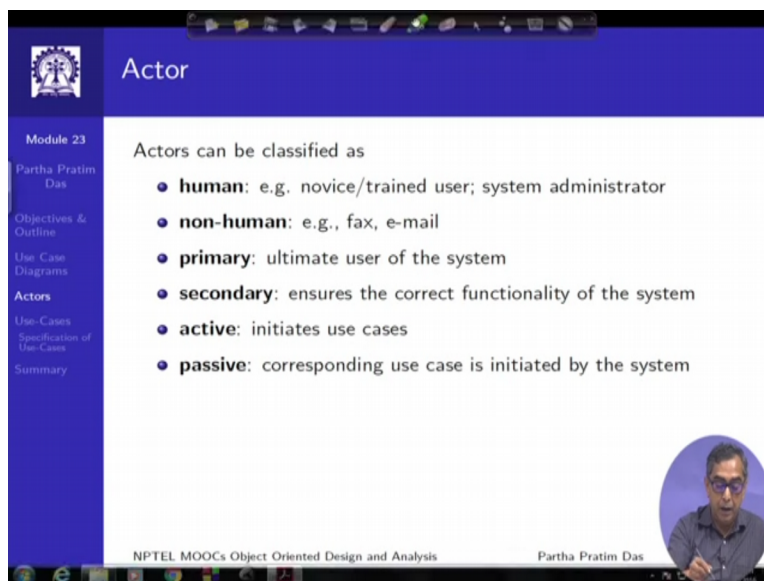
So, the uml diagrams gives various different notations and semantics for defining these components.



First the actor. As already explained actors interface with the system. Actor though the name sounds as if actor is a human being but uml considers actors who are people or human beings or actors who are other systems that is I can get a request from a human being sitting on the terminal working through a gui or I can get a input, I can get a request for service from some other system also. Actors could be non-human.

These are typically shown in terms of typically shown in terms of stick diagrams figures like this. Or they could be level by guillemets also I mean instead of the figure I could just put guillemet is this pair of symbols. We put a name within that and that means an actor.

(Refer Slide Time: 06:39)



The screenshot shows a presentation slide titled "Actor" with a blue header. On the left is a vertical navigation menu with items: "Module 23", "Partha Pratim Das", "Objectives & Outline", "Use Case Diagrams", "Actors", "Use Cases", "Specification of Use Cases", and "Summary". The main content area has the heading "Actors can be classified as" followed by a bulleted list:


- **human:** e.g. novice/trained user; system administrator
- **non-human:** e.g., fax, e-mail
- **primary:** ultimate user of the system
- **secondary:** ensures the correct functionality of the system
- **active:** initiates use cases
- **passive:** corresponding use case is initiated by the system

At the bottom right is a circular video feed of a man. The footer text reads "NPTEL MOOCs Object Oriented Design and Analysis" and "Partha Pratim Das".

There could be variety of different actors or or rather the actors can be classified in multiple different ways. For example, the first-class verification I have already talked of, an actor could be human or non-human. An actor could be primary or secondary, the primary user is kind of who is a targeted end user of the system so in the leave management system, an employee or executive is a primary user. And a secondary user is someone who is required for the correct functionality of the system.

So, system administrator who has to make sure that the proper leaves are credited the leave, records are reconciled and all that is considered to be a secondary user. Then users could be active or passive. Active users are those who initiate a use case and passive users are those who are recipients of the use cases. So, an active user starts applying for a leave, a passive user is like a ehm actor is like a printer who based on a request to print a report would actually print the report.

(Refer Slide Time: 07:50)




# Actor

Module 23  
Partha Pratim Das

Objectives & Outline  
Use Case Diagrams  
**Actors**  
Use Cases  
Specification of Use Cases  
Summary

How can Actors be identified?


- Who uses the essential use cases?
- Who needs system support in order to fulfill the daily tasks?
- Who is responsible for system administration?
- What are the external devices/software systems the system has to communicate with?
- Who is interested in the results of the system?



NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

So, the question again is how can we identify the actors. So, we use a essential use cases who needs the system support who is responsible for system administration, what are the external devices, software systems to communicate with, who are interested in the result of the system and so on. These are the questions you can keep on asking and answers for those will give you the actors.

(Refer Slide Time: 08:24)




# LMS: List of all Identified Nouns: RECAP (Module 17)

Module 23  
Partha Pratim Das

Objectives & Outline  
Use Case Diagrams  
**Actors**  
Use Cases  
Specification of Use Cases  
Summary

Company	Attendance	Leave	Employees
Contributors	Lead	Executive	Manager
Leave Rules	Days	Year	Name
Type of Leave	Period	Absence	Holiday
PL	CL	EL	DL
SL	ML	LWP	UL
Pre-approval	Month	Service	Quarter
Medical	Parenthood	Disciplinary	Administration
Certificate	Certificate	Action	Function
Daily	Personal	Calender	Batch Task
Attendance	Details	Year	
Account	Balance	Designation	SysAdmin
Parent	Salary	Week	List
Privilege	Right	Login ID	Leave Status
Employee Code			

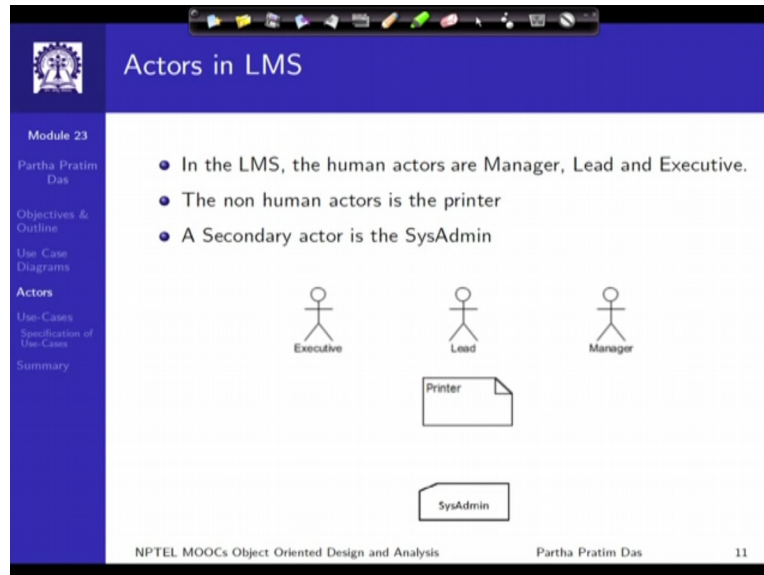


NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

And extending on our earlier approach of analysis to identify objects and classes, I could easily conclude that certainly an actor will be one who is an identifiable class, identifiable object. So, if we refer back to our linguistic analysis of nouns that could give us some good clue in terms of what the actors could be, so we look in here all around and try to find out you know nouns like employee, manager, lead, sysadmin and so on which could initiate some action in the system.

The employee initiates the request, lead initiates a approval process or a revocation process, sysadmin initiates the leave crediting process and so on. So, you can very well understand that there is there is no very standard mechanism for identifying actors but again the linguistic analysis of noun will give you some good clue and then clubbed with that if you do some you know action analysis based on the works you will be able to get a better confidence as to what, who are the people who would need to act on the system and must be actors in the uml diagram.

(Refer Slide Time: 09:49)



So, this is the lms example which I have already discussed. So, these are the 2 different ways so typically you use a this kind of a icon when you know that the actor is a human and you use this kind of an icon when you know that the actor is a non-human. So, these are the actors in the LMS system that you can easily see.

(Refer Slide Time: 10:16)

The screenshot shows a presentation slide titled "Use Cases". On the left is a blue sidebar with a logo at the top, followed by "Module 23", "Partha Pratim Das", and a list of topics: "Objectives & Outline", "Use Case Diagrams", "Actors", "Use Cases", "Specification of Use Cases", and "Summary". The "Use Cases" item is highlighted. The main content area has a blue header with the title "Use Cases" and a white body containing three bullet points. At the bottom, there is a footer with "NPTEL MOOCs Object Oriented Design and Analysis", "Partha Pratim Das", and the page number "12".

## Use Cases

- Use cases represent what the actors want your system to do for them
- Each use case is a complete course of events to be executed in the system from a user's perspective
- Use cases can contain short descriptions course of events in the system from a user's perspective

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 12

Now the next component of use cases diagram and certainly the most important component is the use case itself, that is a use case represents what the actor want the system to do, the required action. So, if an executive is an actor, executive wants to apply for leave. The lead needs to review and accept, approve the leave. So, these are what the actors want the system to do. So, we just do not end by identifying this action is required but you say that a use cases has to give us a complete picture, complete course of events.

For example, if if an executive is applying for a leave, then it's just not applying for a leave, it's the whole course of actions have to happen, you must have a reason for the leave, you must have a enough account balance for that leave, it must block those dates, it must calculate how many days it turns out to be, it may have salary implications and so on. So that complete course of actions from the use case for a particular action. In addition, use cases that that diagrams as you say, but they could have some attached nodes, short nodes to describe what the use case is talking about and what the conditions is and so on.

You would I would like to remind you at this stage that use case is a diagram to be used at the very beginning of the whole activity. So, you are using use case at the requirements phase so when you are working on the use case you do not have any of the other uml diagrams possibly at hand. So, all that you are relying on is your classical natural language descriptions and discussions and so on. So, in one step you may not be able to put everything concretely in terms of the required graphical notation of the uml.

So, you may still need to have some further details, even some some details may not be representable in use case diagram. So those are the things you keep putting down in terms of the notes and attach them with different parts of the use case diagram.

(Refer Slide Time: 12:40)

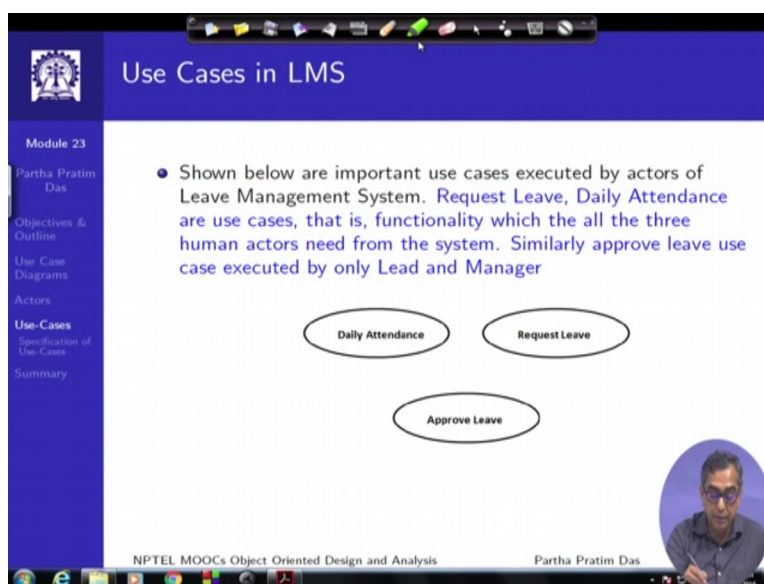
Module 23	Wants	Manage	Work	Report
Partha Pratim Das	Approve	Regret	Credit	Join
Objectives & Outline	Prorate	Cross	En-cash	Paid
Use Case Diagrams	Allow	Send	Need	Become
Actors	Enjoy	Avail	Proceeding	Employ
Use-Cases	Consider	Deduct	Provide	Request
Specification of Use-Cases	Cancel	Check	Export	Revoke
Summary	Debit	Adjust	Perform	Hire
	Fire	Generate	Leave	Can be
			En-cashment	Available
	Can be	Can't be	Can't be	Can't be
	Clubbed	Availed	Carried forward	Clubbed
	Can't be	Accumulated	Proposed for	Join Back
	Continued	Up		
	Doesn't Draw	Can be Revoked	Leave Credited	

Many extracted verbs are in derived forms – so we extract the unique stems

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 13

To identify use cases again I would refer you back to the linguistic analysis of verbs and we can see that the employee needs to approve o I mean a lead or manager needs to approve a leave, regret a leave or revoke a leave, an employee can cancel a leave and so on, can avail a leave. So, these are different actions that are expected in the system and therefore they will very legitimately qualify a candidate to be considered for different use cases.

(Refer Slide Time: 13:23)

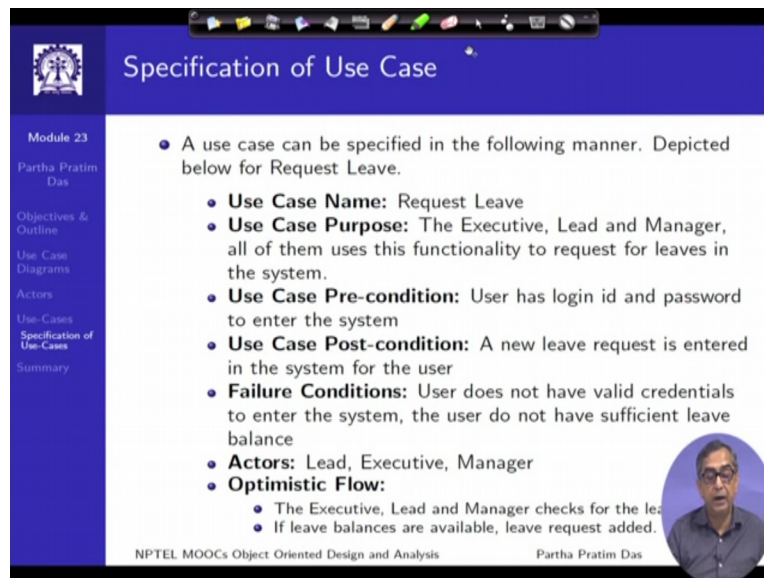


So very explicitly these are some of the request leave, daily attendance are some of the use cases, the



use case is typically drawn in a kind of a oval shape and with the use case name written here. So, this is the shape and this is the name of the use case. So, this is the minimum that the use case must have. It may have other annotations, other ornamentations along with it but it must at least basic have a closed oval elliptic shape and a name to being identified as a use case.

(Refer Slide Time: 14:07)



**Specification of Use Case**

- A use case can be specified in the following manner. Depicted below for Request Leave.
  - **Use Case Name:** Request Leave
  - **Use Case Purpose:** The Executive, Lead and Manager, all of them uses this functionality to request for leaves in the system.
  - **Use Case Pre-condition:** User has login id and password to enter the system
  - **Use Case Post-condition:** A new leave request is entered in the system for the user
  - **Failure Conditions:** User does not have valid credentials to enter the system, the user do not have sufficient leave balance
  - **Actors:** Lead, Executive, Manager
  - **Optimistic Flow:**
    - The Executive, Lead and Manager checks for the leave balance
    - If leave balances are available, leave request added.

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

You remember I have started saying that uml is grossly a graphical language, so everything every concept, every primitive we talk of there is a iconic representation for that. So along with certain textual annotations its primarily this iconic primitive which will tell us what's exactly being represented and going on in the system. Now a use case may be specified in the in the following manner that is this is kind of a possible way that you can specify a use case.

So again, we are at a very early stage of design so we do not have much of the details codified in the uml diagrams. So, in terms of a use case, at least the first thing naturally is obviously required then name of the use case, you must give it a name. it is good that in the notes that if you put what is the purpose of this use case so that will clarify as to why this use case is important or how this use case could later stage be found to relate to other use cases and so on.

The next one is very important is it says that many of the use cases have a precondition that you can do something provided certain conditions have met in the system. Of course, in terms of the system this is a very trivial pre-condition that the login and password signing in is required to enter the system. But you can get a whole lot of pre-conditions in the LMS system that for example to be able to avail of a sick leave. for the basic approval for the or for the application I do not need a certificate but when it



needs to be finally reconciled I need a certificate.

So, this reconciliation need as a pre-condition of a certificate. The post condition is what happens when a request has been entered am sorry when a use case has actually executed has performed and after that what has to happen. For example, you have applied for leave, the leave is approved, once the leave is approved it has to get temporarily locked in your system to show that you have less number of leaves now. Now when you avail on the termination of avail of the leave it must get actually deducted from your leave.

So, the post condition of having availed is that the your leave permanently go down by the number of days you have taken leave. The post condition of approval is it is provisionally locked to a lower number by the number of leave days that you have been that you have got approved. Because at this stage at the approve stage if the leave is revoked by your approving manager or you cancel the leave, then it will again the post condition would be that it should get unlocked and go back, should get credited back to your leave records.

Often, it's a good idea to also document in the specification that the failure conditions, what happens if the pre-condition is not satisfied or if the use case fails to execute successfully. So, what should be the what should happen in those cases must also be documented. For example, the user has an employee has tried to apply for a leave, balance in the account of the employee satisfy but in terms of the clubabililty condition, the leave was not a valid one. 2 leaves were being asked for together, (()) (18:12) together which are not which cannot be taken together.

So, in that failure condition we need to specify as to what happens to the apply leave use case. Trivial to say that actors need to identify as to who all possibly can work with this use case and often people would also like to put a optimistic flow. Optimistic flow is not a fixed flow I mean is not that, this is not will happen but it's kind of we are designing the systems we are designing the system with a purpose so the optimistic flow say that if everything else go right then this is what we would want the use case to do.

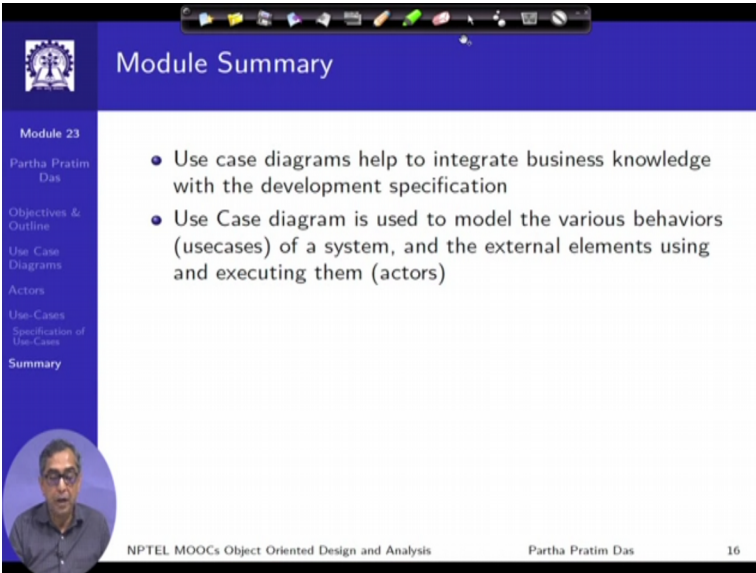
So, these are some of the possible optimistic flow for the request use case we are looking at. So, it's often good to document the optimistic flow because when it later on this use case will go to the analysis phase, the design phase and finally and finally the you know the low-level design will get detailed out,

it will help the designer to understand that this is typically what the system will be used for.

So, the designer at this stage could make recommendations to the test system later on that put a lot of emphasis on these cases because these are the cases in which the user would normally use and test the system. Whereas failure conditions are important to also design tests to decide as to if the system will work in the corner situations. These are the typical these are typical structure of a specification of a use case.

It is not necessary that all of these of course the name is necessary but many of these may be missing in a specification but certainly as much of that can be put in the specification of the use case certainly help.

(Refer Slide Time: 20:18)



**Module Summary**

**Module 23**  
Partha Pratim Das  
Objectives & Outline  
Use Case Diagrams  
Actors  
Use-Cases  
Specification of Use-Cases  
**Summary**

- Use case diagrams help to integrate business knowledge with the development specification
- Use Case diagram is used to model the various behaviors (usecases) of a system, and the external elements using and executing them (actors)

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das 16

So, to summarize on this module, we have started discussing the first uml diagram, the use case diagram which we know is to be used for the requirement specification stage of the SDLC process and in the use case diagram we have talked about the 2 primary components of actor and use case. In the following module, we will continue our discussion on the use case diagram and talk about the relationships that exist between different use cases.