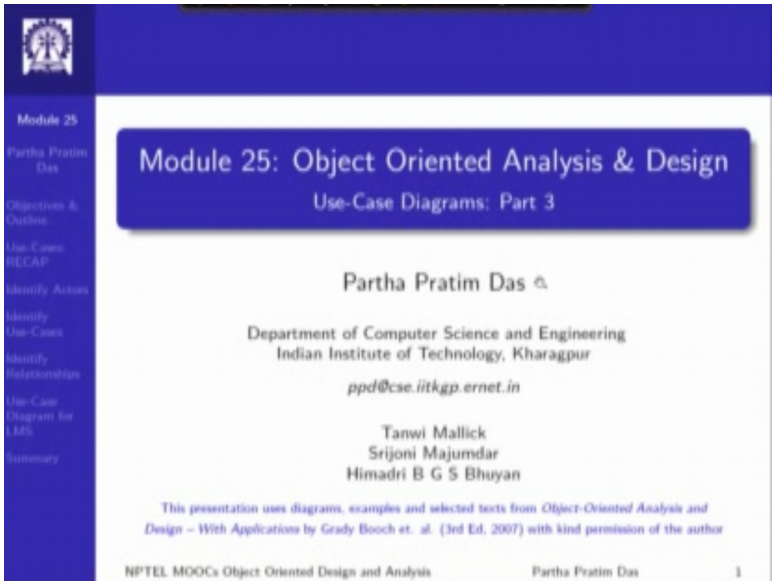
Object-Oriented Analysis and Design
Prof. Partha Pratim Das
Department of Computer Science and Engineering
Indian Institute of Technology – Kharagpur

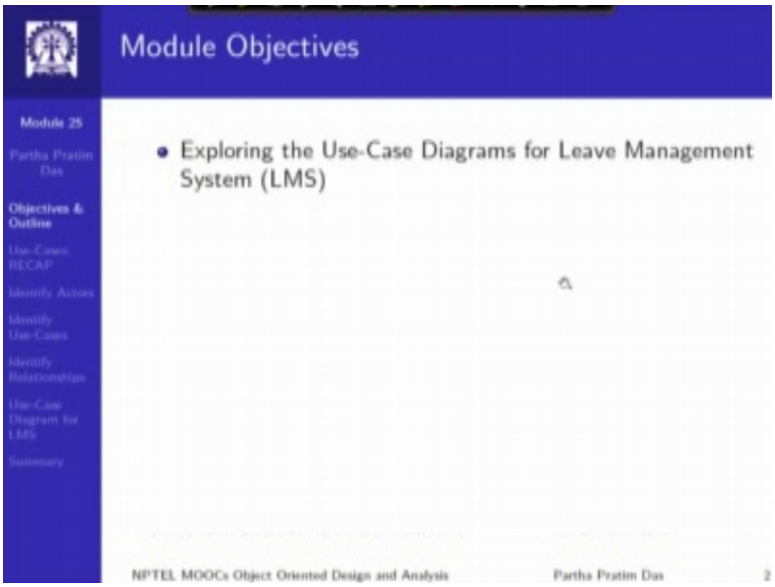
Lecture - 37
Use-Case Diagrams: Part III

(Refer Slide Time: 00:25)



Welcome to module 25 of object oriented analysis and design. We have been discussing about use-case diagrams.

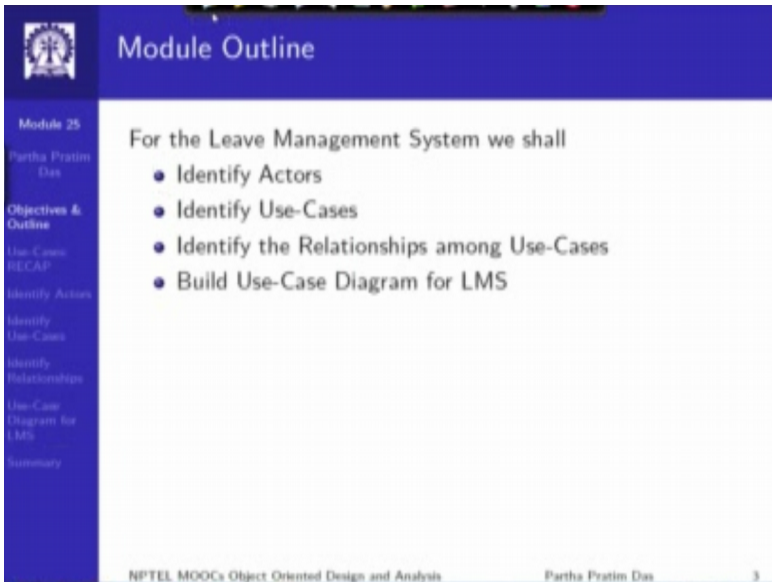
(Refer Slide Time: 00:31)



And we have seen all the major components that constitute the use-case diagram. So in this module we will focus on trying to analyze the leave management system specification further

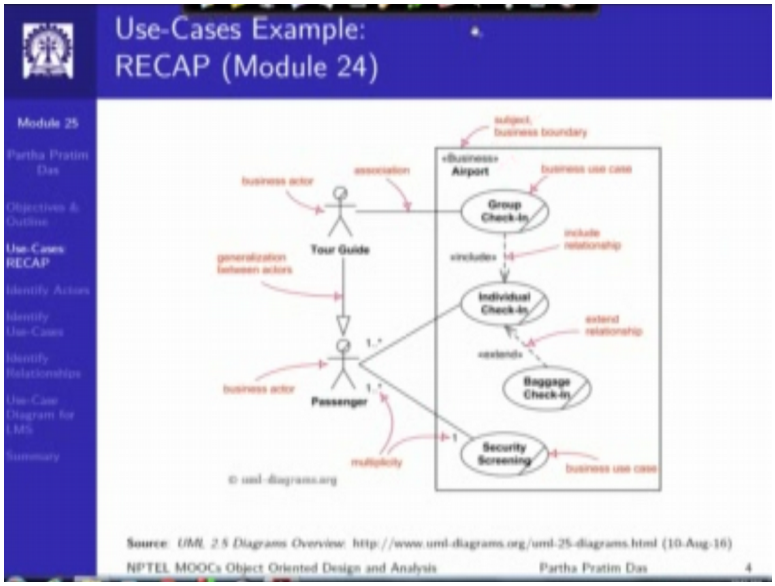
and try to show you how these use-cases can be identified and put in together and we will try to conclude with a first level, very tentative use-case diagram of the LMS system.

(Refer Slide Time: 01:09)



But before I get into that let me also recapitulate some of the major features of the use-case diagrams that we have already discussed. This will be the outline for this module. So certainly as you can see the main actions would be in terms of identification of actors, use-cases and the relationships.

(Refer Slide Time: 01:39)



So first on to the recapitulation. So this is just based on what we did in the earlier module. So this is an use case example, not one which we have discussed earlier. So this is kind of checking in at an airlines counter and the possibilities are it could be a group checking that is if you are going on a tour together or you can do an individual checking, actually it involves

checking in of the baggage and before you can get into the flight you need to have a security screening.

So these are the different use cases that exist. So for each one of them we specify that this is the use-case name. This part is where you can put more additional notes and so on. We identify two actors, this is passenger who is a passenger who is a business actor we say and this another who is a tour guide. So it comes in place when certainly some group of people need to check in for a particular tour.

So the kind of things that we can expect is, one, is include relationship between the group check in and the individual check in. Which means that if 5 people have to check in then necessarily it means that each one of them have to check in individually. So this is an include relationship which say that to be able to complete the group checking it is necessary to complete the individual checkings.

Similarly, we can see that between the individual checking and the baggage checking there is a exchange relationship. Why is it an exchange relationship because if someone is checking in for a flight, then whether or not the person will carry some baggage to be checked in is optional. It is not every passenger, some passenger could just fly with one bag and does not need to do a check in. So this optionality is demonstrated by the extend relationship.

So individual check in at an extension point which is not shown here possibly saying that has baggage kind of will invoke the baggage check in use case. Then there is another here. Some more that we can absorb here is between the actor tour guide and the passenger there is a specialization relationship because any tour guide can also be a passenger might travel with the same group, but a passenger is not a tour guide necessarily.

So a tour guide can do all that the passenger like, tour guide can individually check in, will go through the security screening but this is not possible, that is a passenger cannot do a good checking. So that is exemplified by this generalization, specialization relationship between the actors what we are showing here. Further you can observe that certainly actors are associated with use-cases, so the tour guide is associated with the group checking which is marked by this association link.

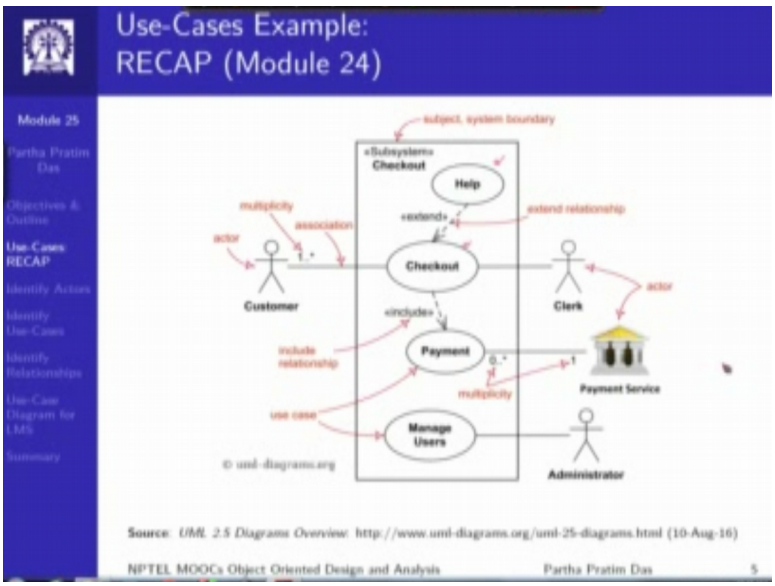
The passenger is associated with the individual checking which is marked by this association link and we can also see that there could be multiplicities at the end, both ends of the association. The multiplicity means that how many different actors can associate with how many different use cases. So here we say it is 1 dot dot star which means that one to any number more than one.

And here it is one. So it says that any number of passengers, at least one passenger has to be there, but any number of passengers can go through security screening. Similarly, in this case of association between passenger actor and the individual checking use case, the 1 dot dot star again means that any number of passengers can check in individually. Here the one is not mentioned, if it is not mentioned then it is meant to be one always.

So this is called multiplicity. What else do you observe? We observe that there is a bounding box that puts all this use cases together. This bounding box is the perceived boundary of the world that we are trying to model and it is often called the business boundary and or the subject of the use case diagram which bounds the whole system and typically subject will have a name to identify this particular business boundary.

And we will find that always the actors will be outside this business boundary and they are not inside the system dynamics, they come and interact with the system to get the kind of services that they would want.

(Refer Slide Time: 06:50)



This is just looking at another example. This is just an example of a typical internet based purchased process. So what we will have, we have one actor who is the customer. There is one actor in case you are doing a counter purchase kind of, so there is another actor who is the clerk. There is an actor who is administrator and there is some actor like payment service and from the model itself you can figure out that while customer, clerk and administrator are human actors, payment service is a non human actor who is involved in this system.

The use-cases, the typical use-case is a check out which is after you have identified the items and put them in our card, whether it is a virtual card or it is a card at the shop. You go the payment counter and you try to place the order saying that I want to buy these things and I am placing the order and I will make the required payment.

Naturally if you do the check out, the other actor associated with the check out may be the clerk, this may be nonexistent if you are just doing an internet kind of transaction in which case the clerk may again become a non human actor, but at the counter the clerk is a human actor. So the clerk has to take part in the check out because she accepts the goods that you want to buy and prepares the orders for you.

Certainly the check out will include payment because you are not allowed to go away with the goods until you have made the payment. So to complete the functionality of the check out use-case you need to perform the functionality of the payment use-case. Now payment assuming that you are making the payment through your credit card or debit card, he will need the payment gateway, the bank and the payment gateway like the Visa, Master, MasterCard these kind of to take part and make that payment happen.

So you can see an association between the payment service and the payment use case through this association. In addition, there could be certain use-case to manage the users to manage the total floor and administrator is the actor who will be associated with managing this. Administrator will not directly take part in the other activities.

So if we look at the associations, customer is associated with check out and the multiplicity is well understood that any number of customers could do check out. Then payment service, one payment service is associated with multiple payments and here you can note that the

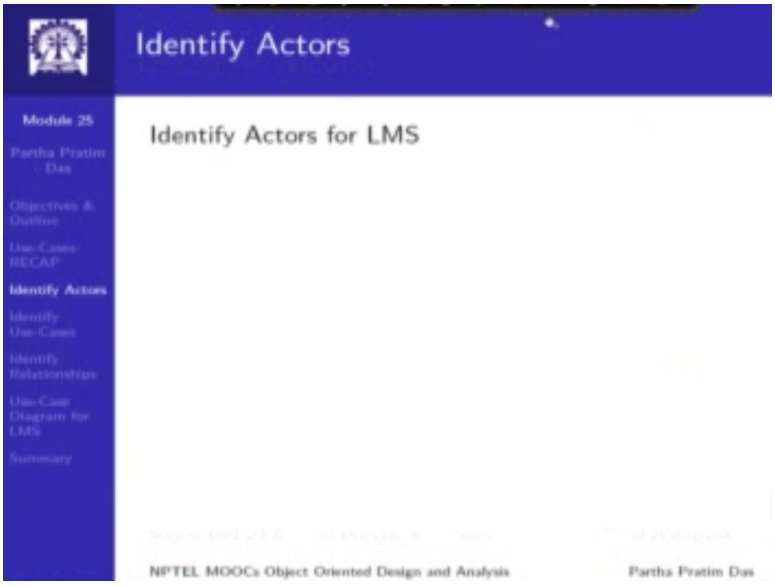
multiplicity includes zero, which means that a payment use-case may not actually use a payment service.

For example, if you are paying through cash then it is just the cash counter. So that is the kind of meaning that the multiplicity can give you. And there is an extend relationship on this site where while doing the check out you might need some help, for example some item may not have a proper bar code and some executive need to be called for the item code and so on.

So you might need different kind of help, but you do not need them as a regular process, you need them optionally under certain conditions and therefore help is an extended use-case for check out. The whole system falls within this system boundary, this rectangle and is given a subject name of check out. So this is just another example of a different system showing how use-cases can be done.

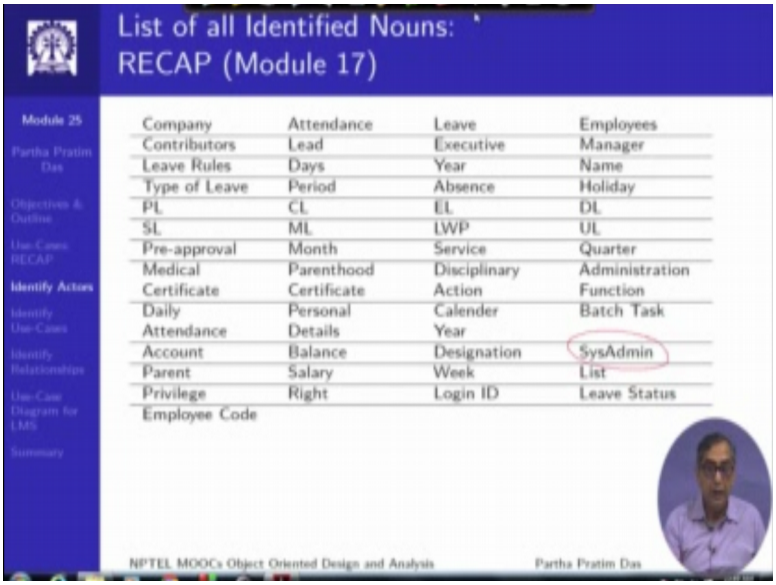
So I just wanted to consolidate all that we have learned in terms of the use-case diagram, in terms of two scenarios airlines check in and in terms of a purchase check out to show you how typically you should be building the use-case diagram for the system that you would want to represent.

(Refer Slide Time: 11:25)



Now let us quickly look over the actions that will need for the LMS certainly to build the use case diagram you need to identify the actors.

(Refer Slide Time: 11:39)

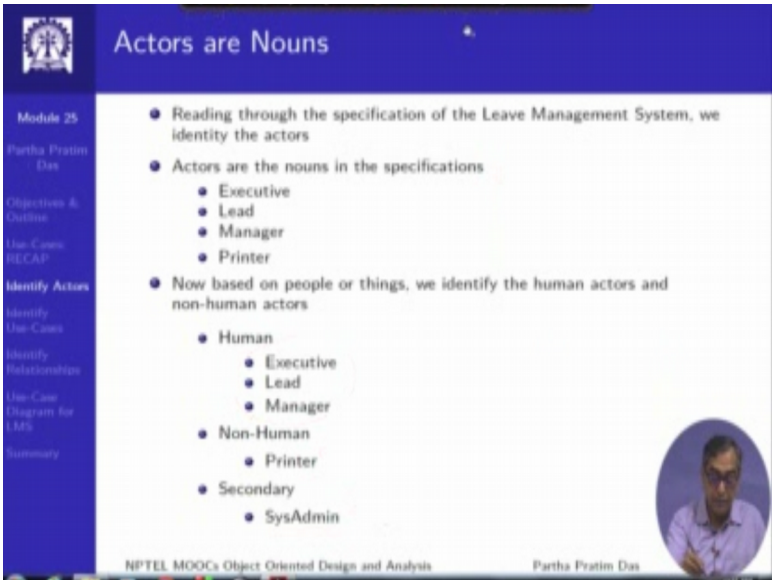


We have already started this activity with the linguistic analysis of nouns. The same thing will be required here, so all that we have analyzed we can look over them and try to identify different actors like manager, like lead, like executive or even like employee and so on. But you will have to keep in mind that all nouns that you identify could be actors for example leave.

This cannot be an actor because the leave is acted upon, you deal with the leave, so it becomes a subject of what you do. But leave is not something that initiates an action which provides you a functionality. So it is not only that you look for the nouns, for example the another could be sys admin who could be an actor, but not only you look for those nouns, but also from that analysis you will have to do a further processing to reason as to who of these could actually act on certain use-cases.

Naturally what we will see subsequently is the identification of actor and the identification of use-case will have to be closely related.

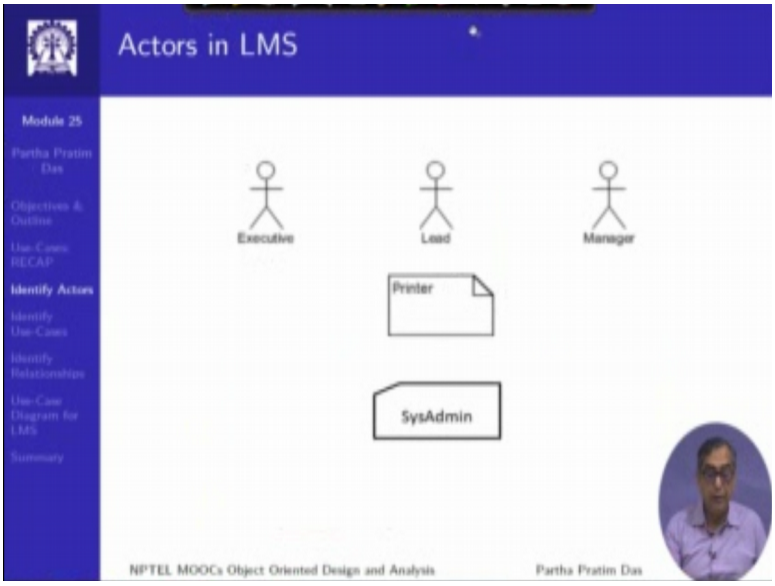
(Refer Slide Time: 13:01)



So if you do this then these are the different nouns as we can identify from this. Now based on these we can have different classification of actors here. The human actors that executive lead and manager, the non human actors could be printer, sys admin could belong here or could also belong here, it depends on what kind of sys admin you have, whether it is just a batch administrator process it is a human being.

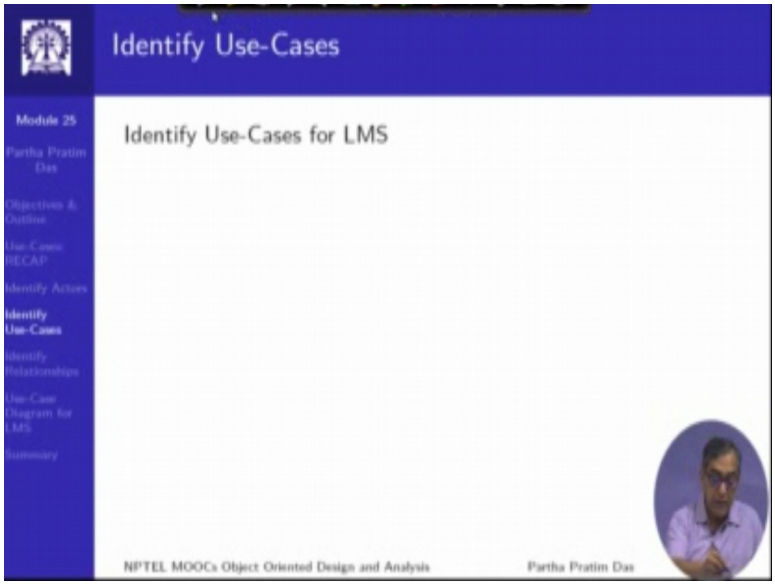
And certainly sys admin kind of actors are secondary because they do not provide the primary functionality of leave management in the system. Whereas these human actors, the executive lead manager are also in terms of our classification are also primary actors. So we could go through this analysis and then again based on the specification and our understanding we could classify all these different designated actors in the system.

(Refer Slide Time: 14:11)



And accordingly we could choose icons for human and non human actors.

(Refer Slide Time: 14:20)



Second is identifying use cases.

(Refer Slide Time: 14:27)

So we will try to identify the use-cases for LMS and use-case is basically as I said is an action, so we will start by trying to identify. If it is an action, then the specification it must associate with some kind of a verb. So we will try to start with a list of verbs that we have extracted from the specification and I am just referring to the list that has just the stem, not all different qualified and derived forms of the verbs.

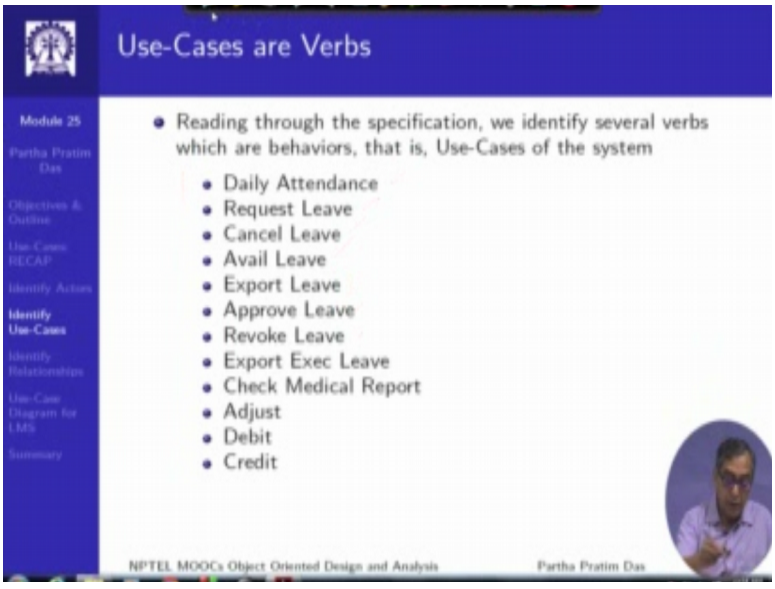
So on this we can see that these are different verbs which approve, regret, request, revoke become use-cases. In fact, you will see that almost all of this verbs in some way or other

represent use-cases, now certainly it is a matter of judgment of the designer of the modeler as to what you want to explicitly say is a use-case and what you just want to ignore. As an example I would just draw your attention to the verb work.

So at the beginning of the specification we have a very small reference that employees work in the organization and of course which we understand also, why are employees there in an organization unless they work. But the question is in terms of modeling the LMS system do we expect to see, use-case called work, I would say possibly not. Even though work maybe the most important use-case for the organization to survive or that is where most of the actors, particularly the human actors would be involved most of the time.

But from the perspective of the leave management system, working is not the matter of importance, working or idling or how some employee, how productive an employee is, are factors which may be important for the organization, but in terms of the lift system, the work may not associate with the use case. So there is a whole lot of judgment and re-understanding revision of the specification that is involved in actually carrying this out carefully.

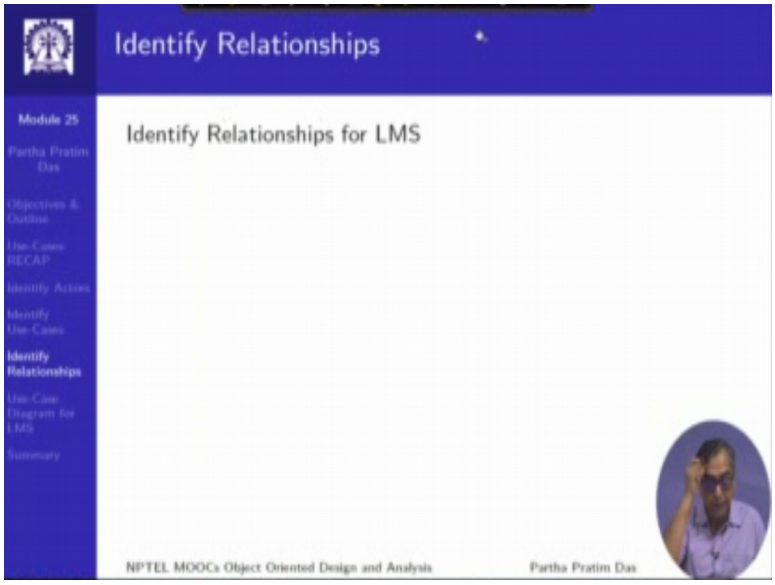
(Refer Slide Time: 16:46)



So if you read through then even after that elimination you will come up with possibly these set of verbs which should be considered as a functionality of LMS system for use-cases like the Daily Attendance process, requesting leave approving leave, cancelling leave, revoking leave, availing leave and checking the medical report, checking other reports and various other supporting use-cases.

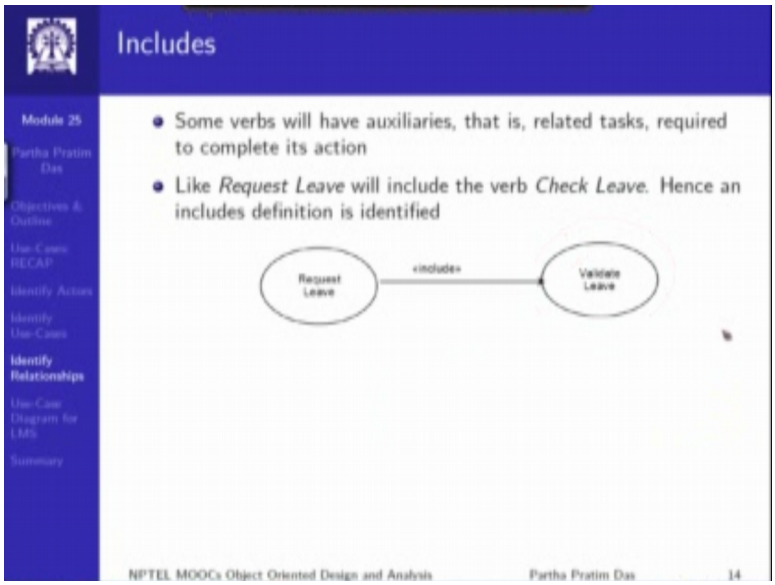
This may not again we can think of that, some are important use-cases like request, approve and so on and some are supporting ones like debit credit which also must happen but it is not primarily visible use-cases of the system. So we will come up with this kind of a list of verbs which could be termed into use-cases.

(Refer Slide Time: 17:52)



Next we need to identify relationships.

(Refer Slide Time: 18:07)



The process of identification of actors and use-cases by analysis of noun and verb and the understanding of the specification, will by itself give rise to a good insight into how the relationship should be looked at. So relationships are primarily of 3 kind, include extend and generalization. So for include, we will need to look at, for each and every use case that we intend to put, whether that use-case is completely self contained.

If it is not, we need to conceive whether there could be other use cases which is required to complete the functionality of this use-case and this analysis of whether they functionality needs to, be completed with another use-case will give rise to the include relationship and mind you, while you do this exercise, it is also possible that for example if we just let me for once go back to the wow used cases, so these are the used cases that we have seen, now in this we do not have a use-case, called say validate leave. I think we do not have this.

We have all this different use cases and if you actually read the specification why we did not have this, because in the specification, this has not been explicitly stated and so when we look for the verbs we did not see a verb like validate or check occurring and therefore we missed that out. But once we tried to look at the functionality and we tried to look at the functionality of request approve, then we find that how can I request for a leave unless that request is a valid one.

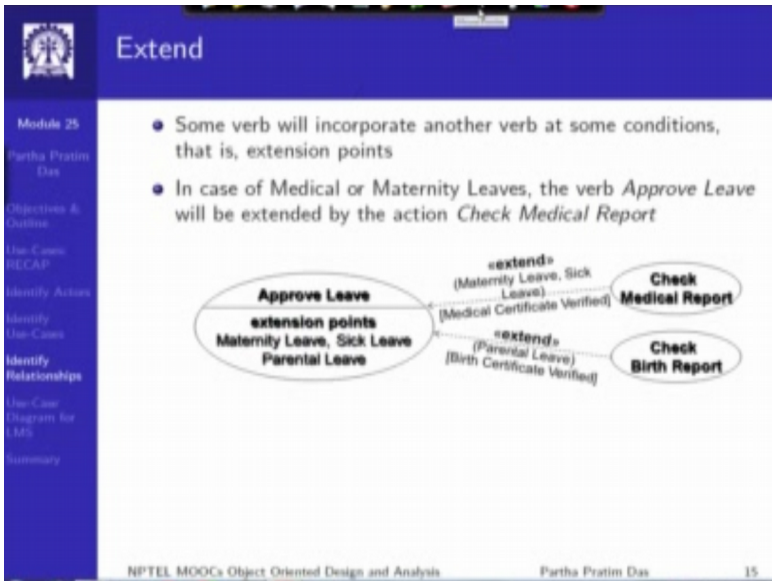
For example, if I am depending on the type of leave, I am applying for example, if am applying for casual leave and if it spans over 4 consecutive days, then we know from the specification that it is something that cannot be approved. If I am applying for a medical leave, sick leave and I am not supported by, valid reports, then it will not be sustained. So there are different kinds of conditions and sub tasks that will need to be done.

So if you request for a leave or if you approve, want to approve a leave, then you will need to check if that leave under consideration is valid or not. So this is a use-case which we discovered in the process of analysis and it is not explicitly stated in those terms in the specifications. And we try to put those as include here, so I would just like to remind you that the include relationship necessarily is the best handle to provide re-use.

That is if you do not, if you identify less number of include relationships because you may not find them explicitly in the specification document, the consequence could be that request leave has to put all the cord, all that action required for validating a leave and the approved leave will have to duplicate all of those again as its own functionality whereas they are the same functionality of validating the leaves of different types.

So the reuse will be enhanced, if we can identify such use-cases which can be included in one or more multiple cases and can improve the re-use and bring a better modularity to the system.

(Refer Slide Time: 22:05)



Second is the situation of extend, which is deciding that in terms of a use-case whether we have conditional behavior. So what we are trying to do, we are trying to say in this case trying to look at approve. So I want to make approve a use-case which is a very valid justifying one that is a basic functionality, so that will be a use-case.

Now I think if somebody has to approve, I go back to the document, if somebody has to approve, can the approval be done blindly. Now mind you this approve leave has already included a validate leave, so the system has already validated that the leave according to the basic structure of leave duration, leave type, availability of leave to that particular employee and so on has already been changed.

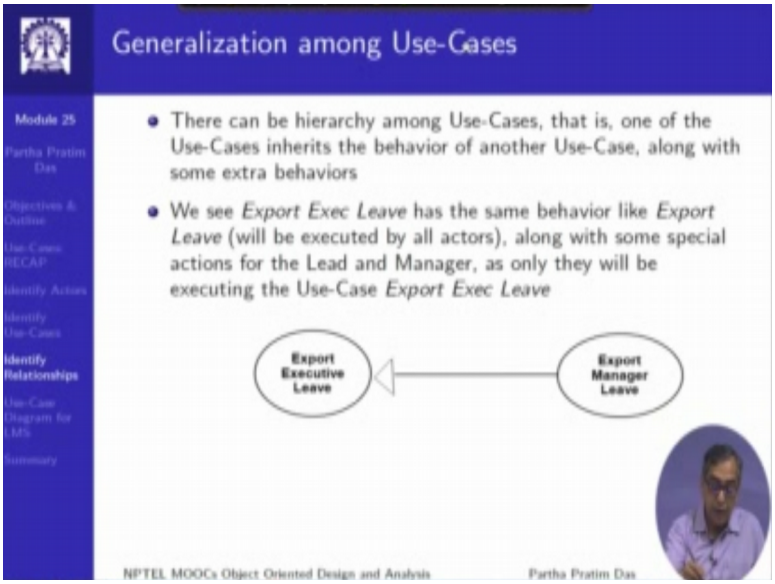
So is it that for all leave, if it is a valid one, then the lead can simply blindly go and approve it provided the lead is okay with the absence of the employee during that period or there needs to be further use-cases to be used. And we find that some of the leave have other conditions attached and that is the situation of medical leave, the sick leave as we mention there, the maternity leave, parental leave and so on.

So if I want to talk about approve leave it will have to for the case of medical leave or sick leave, for the case of parental leave it will have to involve further conditional processing

which is not applicable to approve leave of other kinds and that is the kind of situation which gives rise to conditional optional use cases and that is exactly the extend behavior.

So in terms of identifying the extend behavior you are specifically looking at, what are the situations where a general processing or a general use case action, may require very specific sub use-cases to be identified and used as extension on a conditional basis and that is the example. We have discussed this particular example earlier also, but that is the process that you do. The identification of conditions, the process that you do to extract the extend relationship in the model.

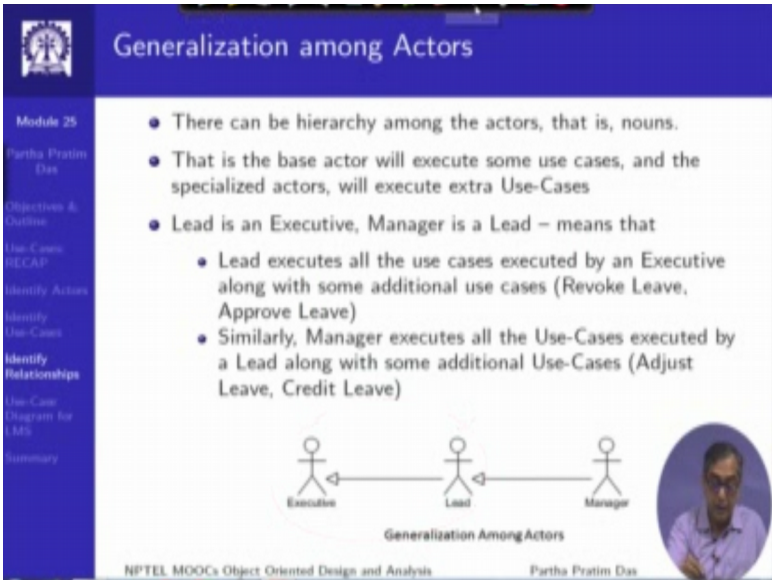
(Refer Slide Time: 24:41)



Generalization is certainly everywhere. Generalization of use-cases will look for, that once you are have identified first should be that you try to identify as many use cases as you can and then you try to see that between different use cases if there is total subsumption or total commonality of behavior as you may have in terms of exporting the leave for an executive and the exporting the leave for a manager.

You see that this behavior of this use-case is totally subsumed in the behavior of the export manager leave or in other words export manager leave needs to possibly do few things extra but otherwise it is like the exporting security leave and like in the way we try to establish the generalization specialization between classes in a very similar way we try to establish the generalization specialization between use-cases and we can build those up in terms of the system.

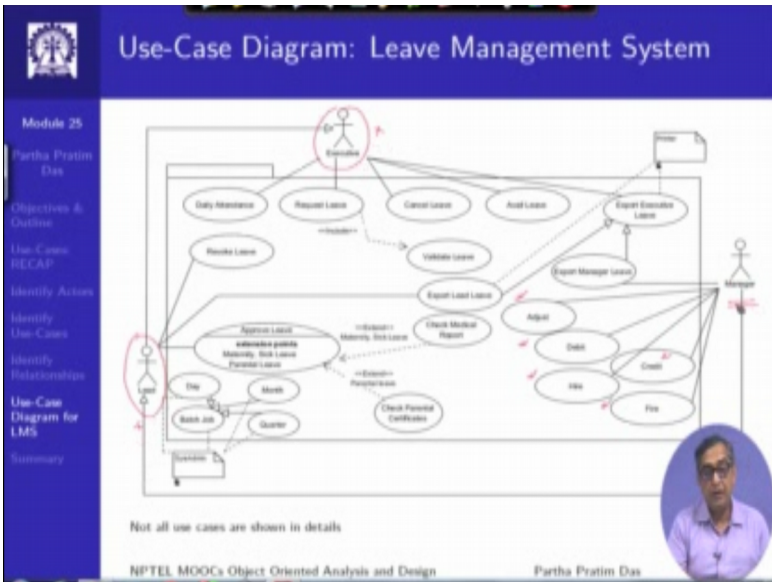
(Refer Slide Time: 25:48)



Generalization among actors are very similar. You will need to identify what actions, what use cases the actors are associated with and if we find that the use-cases or actions for an actor, say an executive is completely covered by the use-cases that lead employee, the lead actor has to perform, then he can say that lead is an executive, or lead is a specialization of the executive.

Mind you, whenever we put this generalization specialization relationship that anything that this actor will be associated within the use case, this actor would be able to perform that. So you must, if there are exceptions then it will, this generalization will not work. So you will have to be careful about doing this.

(Refer Slide Time: 26:48)



So using all this, combining all these techniques of identifying actors, use cases and relationship we finally consolidate for the leave management system. So here I am trying to show, the use case diagram with the actors the executive lead and manager and we can see that there is a generalization relationship amongst them and there are other non human actors as well.

And with that now all the major use cases have been put in place, for example an executive will go for daily attendance, can request for a leave. If the leave is requested, then request use-case will include the validate leave, so that the leave can be validated. The executive can cancel and avail leave and so on. And certainly export the list of leave.

Now what happens is as we have seen that lead is a specialization of executive which mean that the lead will also be able to provide daily attendance, will also be able to provide request leave, will also be able to cancel leave and so on, but we do not actually need to depict them on the diagram, because since a lead is an executive, the lead will, through the executive the lead will be associated with all the use-cases to which the executive is associated.

So what we do, we just here try to associate the use cases that the lead can perform, but like revoke leave or approve leave, certainly the executive in turn cannot be associated with this used cases. So we have the approve leave, and as we had earlier discussed the approve leave had couple of extension points, so it extends and it needs to check for medical reports in case of SLML it exchanges and check for parental leave condition and parental certificate for this.

So all these are being put together and then if we look at the manager, we find that the manager also is a lead. So which means that the manager can actually perform all the, can be associated with all the use-cases like approve leave, like revoke leave, as what the lead can do, and since lead in turn is associated with request leave with cancel leave and so on, the manager also will be able to.

So manager basically kind of get associated with these use cases to which directly in the diagram the executive is show associated in two levels, since lead is an executive, lead gets associated with request leave and since manager is a lead he in turn will get associated with the request leave. But specifically manager can also associated with number of use-cases like

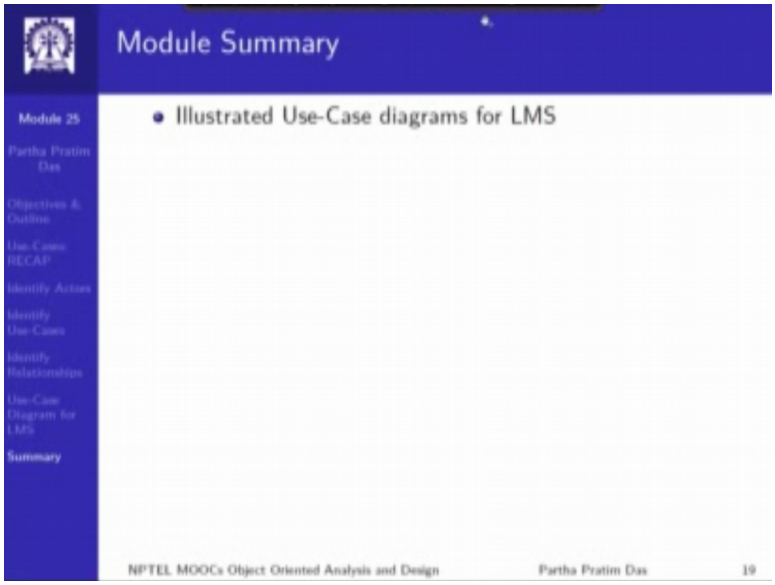
adjust, debit, leave, credit leave, hiring firing of employees and so on, which neither the lead not the executive will be associated with.

So these are the exclusive associations of use-cases with the manager. Naturally manager has separate use-case to exports manager leave. Lead has separate use case for that. And both of them are specialization of export, executive leave because executive lead does not lead to take care of the reporting employees where as the lead and manager cases need to drive those.

So this kind of puts the different pieces that we are doing together, But I have mentioned here, please note that old cases in old details are not given in this diagram that would have been very cluttered in one slide. Later on we would like to publish the more complete solution of use-case at other UML diagrams on the LMS system.

So you will be able to see more detailed design at the point of time. Mean while I will urge that you follow the LMS specification and work out., the use case diagram in full depth. You can really appreciate how the modeling is happening and when we finally, later on when we published the solution we will be able to compare and check whether you have been on track or you have been different.

(Refer Slide Time: 31:56)



So to summarize, in this module we have tried to take all the different features of the use-case, diagrams and suggested outline a methodology, to extract information appropriately from the specification and populate the diagrams for the case of leave management system which you should use for your practice. And we have also published the assignment

management system, on which assignments have also been given. So you can also use that and start practicing on the UML diagrams.