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

Lecture - 31
Identification of Classes, Objects and Relationship in LMS (Contd)

Welcome to module 19 of object oriented analysis and design. For the last two modules, we have been trying to apply different techniques and assessment to explore the design of leave management system. So this module primarily continues on that and I would again repeat that please keep the PDF of your LMS system handy please refer to that regularly as you listen to this module.

And also frequently refer to the tutorial video where we have provided a client vendor interaction on the LMS system and keep paper and pen handy so that you can frequently pause this discussion and actually work out the details of what we are outlining here. In this module, we will focus on primarily the extraction of collaboration, modules and hierarchy. So as you have studied earlier in the earlier modules we have discussed the core fundamental items of object oriented analysis and design of the object and classes.

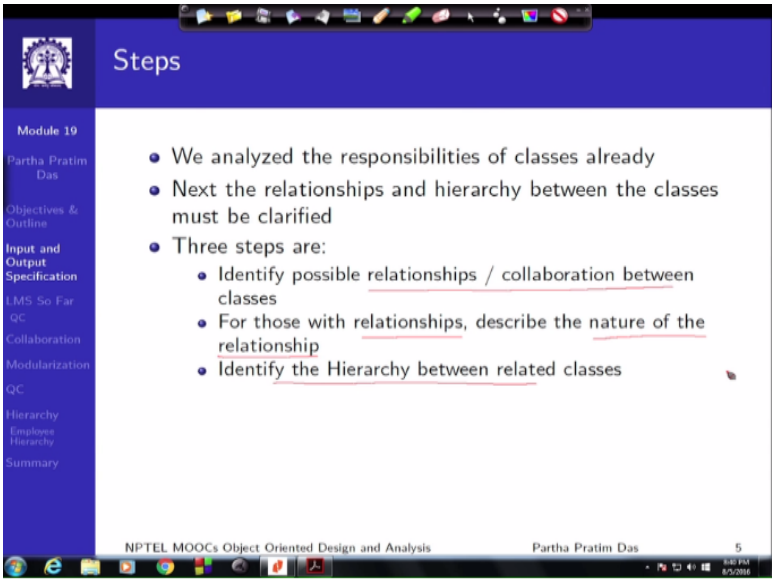
And next comes their relationship. The relationship defines how the system can be organized, what are the different structural and behavioral properties in the system that can be observed. So in the last two modules we have been focussing primarily to extract the information of the classes and object along with that attributes and responsibilities and now we will try to focus more on the possible relationships that different object, different classes may have in this system and we will use two major source.

One is certainly going back to the specification that we have and also we will try to now refer to what we have already designed and try to look for refinements in terms of the relationship that we are trying to find out. So in contrast to the earlier two modules where we were using the specification and trying to find out the classes attributes and responsibilities. In this module, we take these classes attributes and responsibilities as primary input.

But that does not mean that we forget about the specification document because there is still a lot of information, the domain information there which need to be found out, but primarily

from the classes their attribute and responsibilities in this module we are trying to extract the collaboration the modularization in the hierarchy. So these are the main output that we would like to generate from here.

(Refer Slide Time: 03:53)

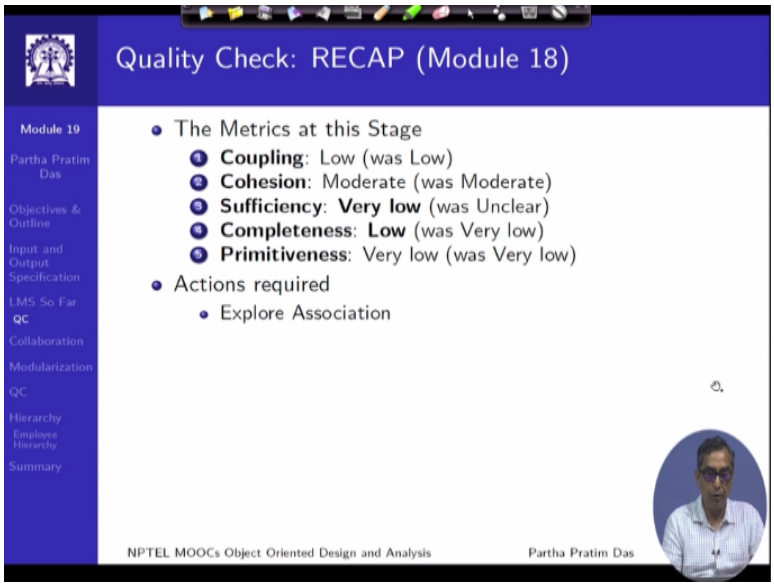


Now just a quick recap of the broad steps as we have already analyzed the responsibilities of the classes like we did in the last module. The next will be the relationship and hierarchy between the classes that we need to establish. So the main three steps are the possible identifying the possible relationships and collaborations between classes and for those relationships we need to describe the nature of the relationship.

And we will do some bits and pieces of that here, but more and more as we will go through the OOAD design process we will see that identifying relationship and describing their nature or characterizing them are key components of the design. And finally the third step would be to identify the hierarchy between related classes and particularly in the next module we will try to show how this hierarchy can be really developed well using the study of different relationships that may exist.

So before we start discussing on this module the design part let us recap what we have in our design by now. We have about 15 classes which we did in module 17. We have identified the attributes for this classes partly you have completed them already I hope and in module 18 we have completed that task of associating the classes with attributes and responsibilities and worked out two samples for employee and leave and left it as an exercise to complete for you all.

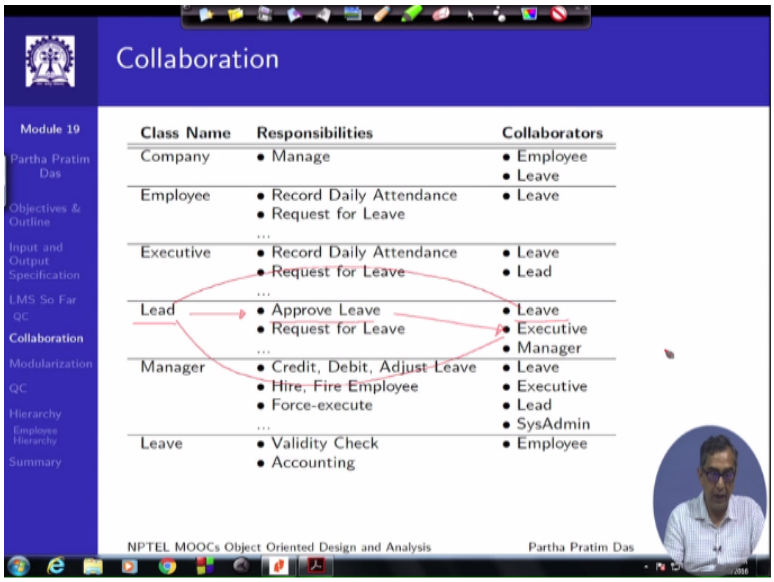
(Refer Slide Time: 05:45)



And at the end of that process our metrics at this stage just to remind you that we are regularly measuring the metrics in understanding how well we are doing. So in the metrics at this stage stand at this level the coupling is low, the cohesion is moderate, the sufficiency is very low. We just have no idea about the whole design.

So we certainly cannot talk about what is the minimalist design the completeness is low because we have just completed the basic round of classes and responsibilities and the primitiveness is also low because we have not done much in terms of identifying the primitives.

(Refer Slide Time: 06:36)



Now let us first talk about the collaboration. So what the collaboration does. Collaboration is

kind of a I mean you can say it is a kind of re refactoring problem for example let me for once again go to the specification document. So let us look at some of these say companies wants to manage the attendance or leave of its employees. So if we look into this manage aspect of this is defining relationship.

This is defining a responsibility for the company then certainly cannot the company disburse this responsibility of management if the company has to disburse this responsibility of management and certainly the company need the employees. So we say that these are there is a collaboration between the company and the employee in terms of the management that happen through them.

We can say that for example if we look into some others then we can say that every executive reports to lead certainly there is a collaboration here which is through the reporting relationship the responsibility. More importantly executive or let say the lead approves leave for some executive. So approve is a responsibility of lead and what does the lead approve? The lead has to approve leave.

So there is a collaboration between the lead and the leave. So this is kind of a factoring problem or you can say refactoring problem where you look into the different responsibilities and try to identify that if there is some agent on the other side of that responsibility. If such an agent exist then we will say that there is a collaboration between these objects or these classes.

So if we come back we can say that a lead this is what I was referring to a lead can approve leave and what does it approve his leave. So the collaboration is here. Now there is a different kind of collaboration in the approve leave is the lead approve leave and for whom does he approve the leave is the executive so there is a collaboration between the lead and the executive.

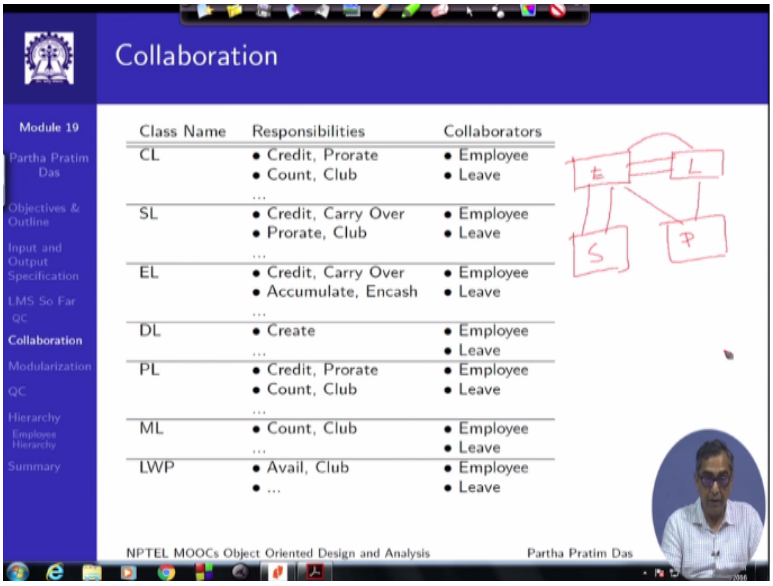
And identifying collaborations is basically again minding through deeper in terms of the specification to say that what are the objects, what are the corresponding classes that would be related through there mutual responsibilities. Managers collaborates with executive through the process of hiring, manager hire executive. Manager would collaborate with the lead in terms of the usual approve leave kind of which is not listed here, but that will also be

here so approve leave kind of responsibility that the manager has.

So in this way we try to identify different classes and their collaborative and again please try to follow similar kind of technique that we were doing in the linguistic analysis also they try to look into every class all responsibilities for every responsibility you try to find out what are the recipients, what are the other actors that exist and put them as collaborators and once you have done that you will find that again there will be lot of repetition.

And you will be able to then refine it to actually find the minimal set of collaboration that need to be in place.

(Refer Slide Time: 11:28)

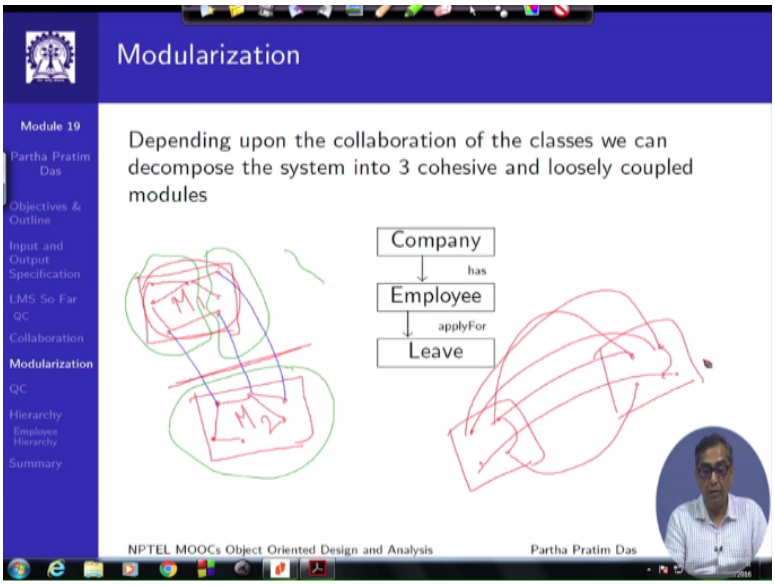


So this is just some more extension if you actually look into the earlier here primarily the first few, primarily where the different kinds of employees whereas in the next these are primarily different types of leave, these are the different categories of leave and their responsibilities and how they collaborate with other classes and you can certainly see that all of these categories will have lot of community in terms of the collaboration and this community is what we will see, we will try to leverage further when we refine that design more and more.

So collaboration if you ask collaboration by itself may not result into a specific aspect of the design, but if we understand the collaboration then we understand that if I have employees here, if I have leave here, if I have system admin here, if I have printer here and so on we have not listed the printer, but then we can see what are the different types of collaborations that may exist between these different classes.

And based on that we can find out how coherent they are, what kind of coupling between them exist and that give a good clue in terms of modularization and all those as well. So this is the basic approach and I would again request you to pause and try to identify as exhaustively as you can all the different collaborations that you see in the LMS specification.

(Refer Slide Time: 13:16)



Modularization is another exercise where you start using the collaboration information more strongly is where you try to see that what would you like to put in a model if we put it differently how do we like to modularize if you remember then we have said that if I have two modules that we have talked about this sometime back then we look into the interaction between different objects in the modules these are the different interactions between the objects in the module.

And we will look into the interactions across the module. So we will say that our task our objective is to define the boundary based on these interactions based on all these what else could be a third colour that I might use maybe this one. So based on all this our intention is to decide should we modularize like this, like this and like this or should I do it differently so that the intra module that is the intra module the traffic that is happening within this.

The collaboration that happens within the module and the once in the blue which are basically inter module reach two extremes that is intra module you would expect lot more of traffic, high density of traffic whereas inter module across the modules we will expect very low density of module or traffic. So if we can achieve that then certainly we will say that we

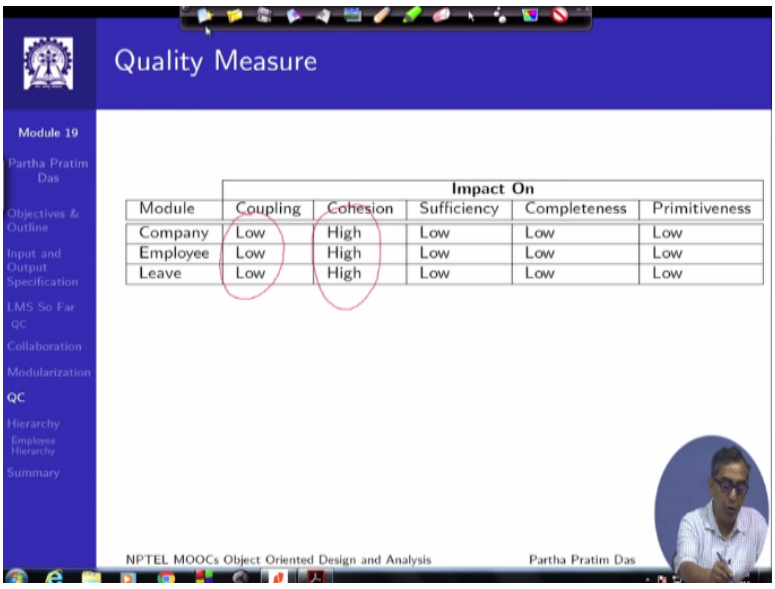
have been able to modularize value.

If we do the other extreme that we did have a module here, we have a module here. We have couple of objects place like this and say this is the interaction, this is the collaboration that exist between them and there is if this is a scenario that we see then we will not say that this is a good modularization possibly we should have put these two in the same module and not across the module.

So this is just to understand that collaboration this information of collaboration would give us a very good inside in terms of how should we modularize. LMS is a relatively simple system so we see primarily the first level we already see three modules emerging one is the company and that module will interact with the employees in terms of companies has employees and then employee module in interact with leave in multiple ways so which can be abstracted in the sense that these are apply for or you know avail of leave and so on.

Of course when you try to complete the exercise you will find that these are not the only modules in the system there would be other modules like your basic accounting maintenance has to be a login module which deals with the login and authentication related activity. There were possibly a backup module to keep backup of the annual leave record and so on, but your basic analysis of the collaboration information which you got from the analysis of responsibilities will give you a initially a good clue in terms of what could be the modules in the system.

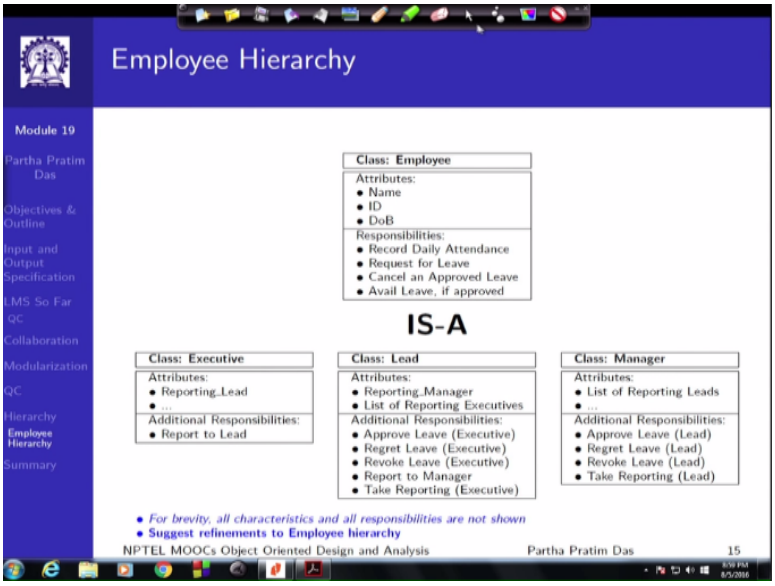
(Refer Slide Time: 17:12)



So at this stage we might take a quick another look into the quality measure to see how we are doing. Naturally the cohesion is improving, cohesion is becoming high we have been able to kind of put the employees together in a module the leave together in a module, the analysis of the collaboration has allowed us in terms of doing that. Company of course is a pathological example because that just one class so it is coherent by itself.

Coupling we still do not know much, we have not seen the coupling aspect though we have the collaboration information, but we have not seen in terms of what the hierarchy could be so this is still low and the other parameters are low because we are still to complete the whole aspects of the design as yet.

(Refer Slide Time: 18:06)



So next let me just start off with the analysis of hierarchy and this is a possibly you saw this earlier also in couple of different other context, but we will say that the collaboration and the responsibilities give us a lot of clue in terms of the hierarchy that we at least should have at the first level. So if you look at the collaboration pattern will tell us that these four classes are strongly related that has also been collaborated by the fact that we found them suitable to be put in the same module.

Now the question is in that module, in that employees module at this classes are they going to remain independent standalone or are they going to be related through some specialization, generalization relationships. So what necessarily help is again going back and looking at the responsibilities, analyzing the responsibilities and if we do that we find that some responsibilities like recording daily attendance, requesting for leave, cancelling an approve

leave for self, availing a leave all these are kind of are found in the employee and actually found in all of these specializations, potential specialization I should say that exist.

So once we see that commonality we start sensing that there is ceratin way to put them in a hierarchy, but of course just be identifying this we do not know if the relationship should be like this or the relationship should be like this that is we cannot say whether executive is an employee or an employee is an executive all that you can say that they share a lot of things that might make sense to test the waters to see if one can be a specialization of the other.

So in one part the identification of hierarchy means that you identify responsibilities, you identify attributes which are common between multiple different classes which possibly by now you have put in the same module. So another would be to looking at the attributes and if you look at the attributes you will find that all of these 4 different classes actually have these attributes of ID which is employee code, name, date of birth and so on.

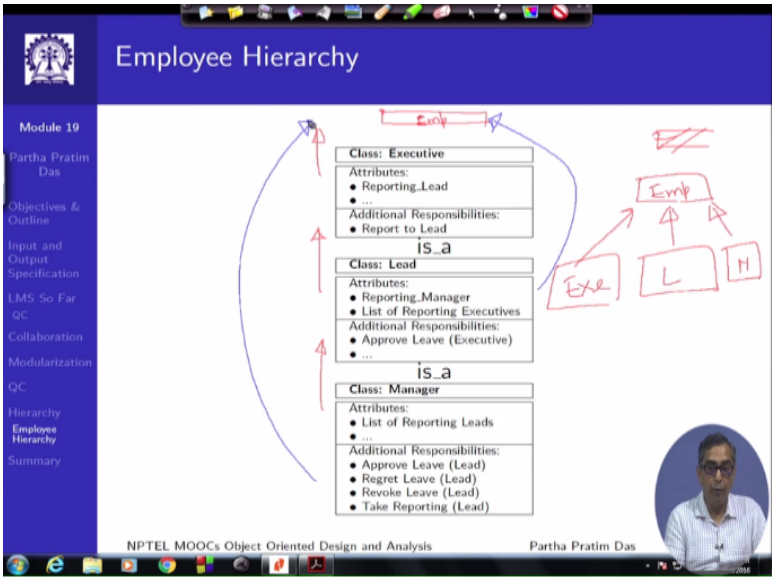
Next to do on the direction of possible specialization you will need to look at where do they differ so this was a common part, common responsibilities and common attributes. Now you have to look at where do they differ why does an executive differ from in general from an employee is there an executive reports to the lead. It is not explicitly said, but this is where you will do some extensional logic that if the executive reports to the lead then there has to be some attribute for an executive which is a reporting lead.

Remind you if you go back to the document look at a list of nouns and so on you will not find this explicit, but this is where we are making certain deductions from the relationships, making certain deductions from the possible collaboration that if executive reports to lead then executive will have to keep the information of that collaboration object which is a reporting lead.

Similarly if we look at the lead then actually we start finding that though there are lot of responsibilities which are common between the employee and the lead, but there is a lot like approving leave, regretting leave, revoking leave and so on which are additional for the lead. So in this way when we analyze all these four classes we find that this class the employee class has kind of the common minimum staff which all other classes have.

And therefore we agree or we decide on an initial specialization where you say that executive is an employee, lead is an employee and manager is an employee. Naturally we do not stop here we would once we have identified this then our effort would be to see can the quality of this specialization we improve this is just one level and certainly therefore it is wide in space.

(Refer Slide Time: 23:29)



Now if we analyze further and now earlier we were trying to see that employee was a common minimum so we are trying to see between employee and each of the other three classes. Now we try to look at once we know that we have a basic structure which is employee, executive, lead, and manager once we have that then we try to see are there commonalities that exist between these specialize classes.

And when we start doing that we do find a lot of commonality between the lead and the manager both of them can for that matter approve leave, both of them can regret leave and so on. Similarly there is commonly between the executive and lead as well both of them can apply for leave. Actually all three of them can apply for leave and so on. So if you do a similar analysis of the containments of responsibilities and the collaboration you will soon come to the conclusion.

Instead of just each one of the employee types being a specialization of the employee possibly a more refined model should have a multi layered structure where you say that the manager is a lead because we find that most of the things if not everything exactly that a lead can do a manager can also perform. When we say lead is an executive because anything that the executive can do the lead can always do.

And finally we preserve the original hierarchy that we are created that an executive is an employee. So earlier we had let say the employee was a specialization for executive for lead and for manager and now that has just become multi layered. So keep this in mind while we do this refinement it will be quite bit if you actually violate the original hierarchy that you had.

So this hierarchy said that lead is an employee so here by transitivity you will also have that lead is an employee. So yo do not explicitly show that because you say that lead is an executive and executive is an employee and therefore by transitivity this holds. Similarly manager is an employee is through the transitivity of all this relations. So when you do this kind of a refinement by exploring more information from the specification, from the problem domain.

You should be careful that your original design aspects do not broken, do not get violated if they do then you will have to review the design very carefully to see whether it is getting violated because what you had assumed earlier possibly based on partial information what had some wrong notions, some misconceptions so you are correcting it now or whether in order to refine you are actually making new mistakes in that design.

So this is the broad way of designing on the employee hierarchy so if we show it in the final form this is how it looks and along with that there are different other relations that they perform we will talk about this more when we talk about the relationships in general. So to summarize this has been the third part of our exploratory exercise on the design of the LMS system.

And in this based on the classes attributes and their responsibilities we have first tried to identify the collaborating classes the collaborators and then we use the collaboration information to create a basic level of modularization in the system and finally using the modules as well as the collaboration and responsibility information we have tried to generate a basic hierarchy of employees and a little refinement of that and as we move into the next module we will show further depths of refinement of hierarchy.