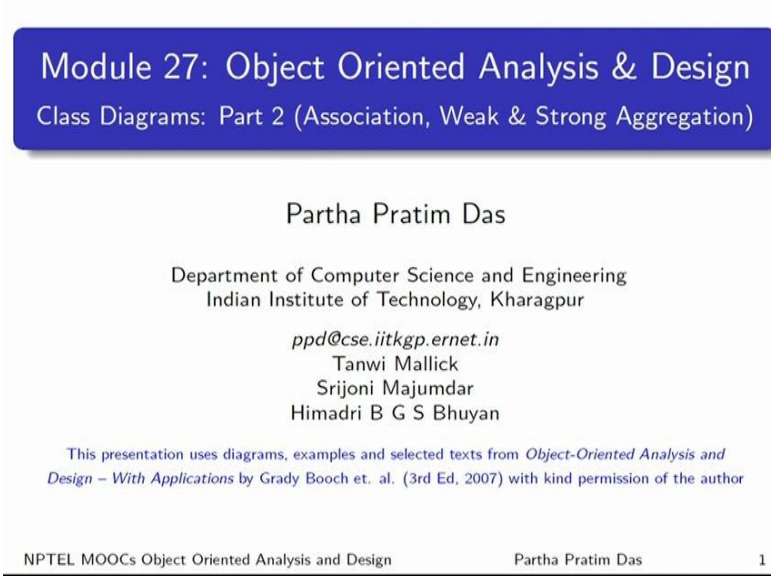
Object-Oriented Analysis and Design
Prof. Partha Pratim Das
Department of Computer Science and Engineering
Indian Institute of Technology - Kharagpur

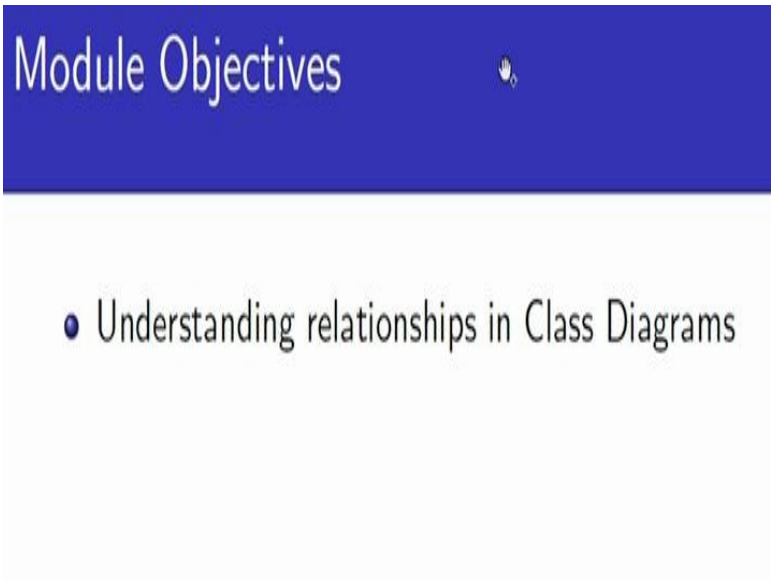
Lecture – 39
Class Diagrams: Part 2 (Association, Weak and Strong Aggregation)

(Refer Slide Time: 00:21)



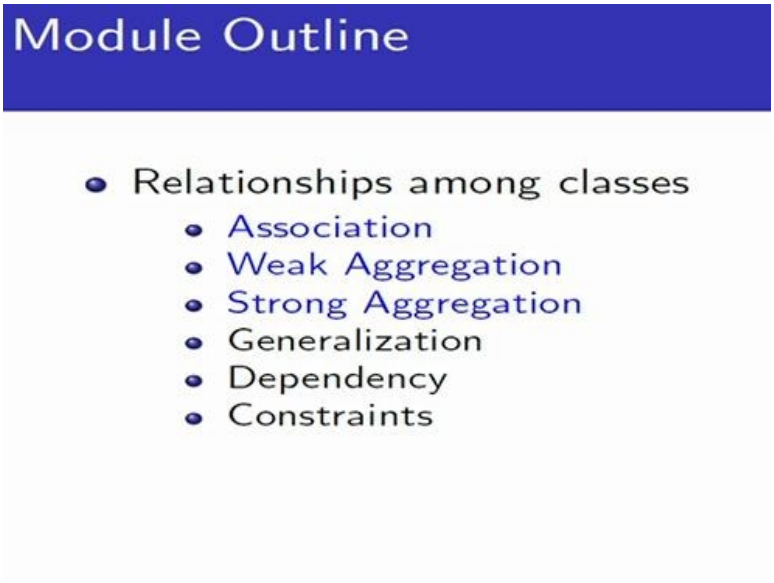
Welcome to module 27 of object oriented analysis and design. From module 26 that is the last module we have started discussing about class diagrams and in that we have introduced the notation for a class, abstract class and the notation for properties and operations. And we have seen different variations that are possible.

(Refer Slide Time: 00:50)



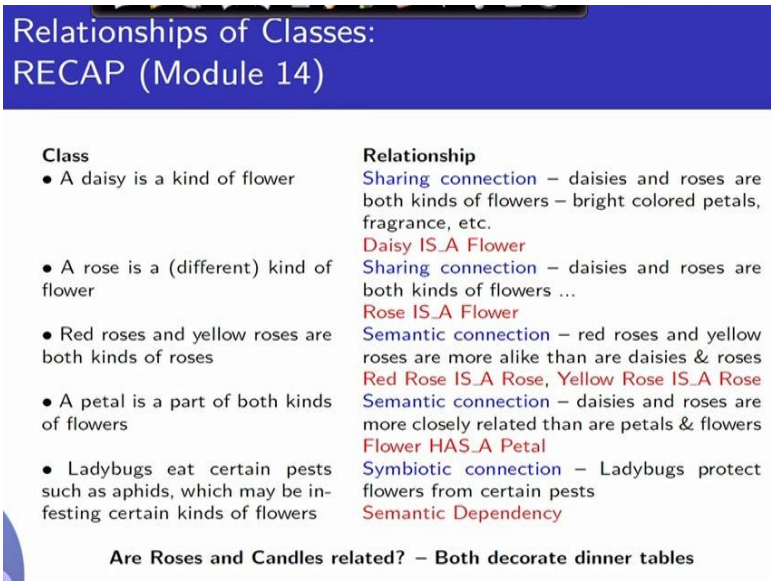
In this module and the next we will discuss about the representation relationships in class diagram.

(Refer Slide Time: 00:55)



So we will introduce variety of representations and since this will be discussed in two modules one after the other I have highlighted the ones that will be discussed here association, weak aggregation and strong aggregation. The remaining relationships will be discussed in the next module.

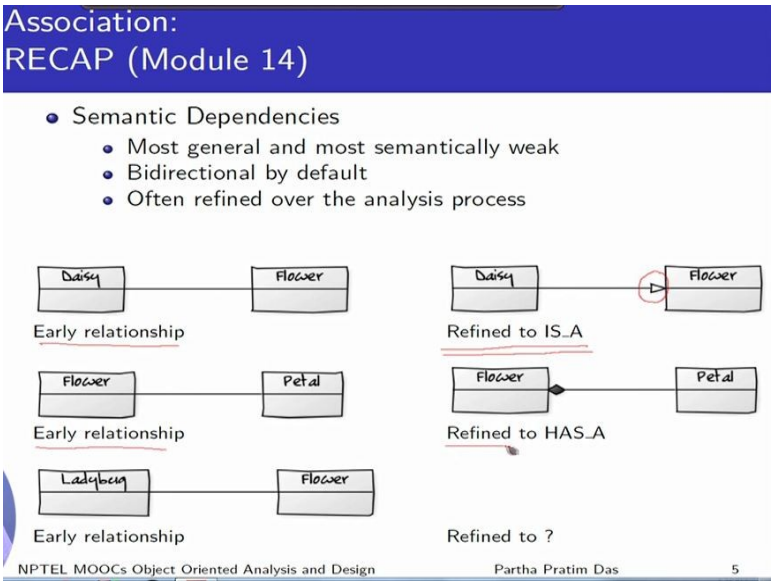
(Refer Slide Time: 01:22)



So just to recap this is referring back module 14 where we talked about relationship among classes with this example of Daisy, Rose this kind of flowers. So daisy is a kind of flower we saw is a sharing connection which leads to IS_A relationship. We saw rose is a different kind of flower similar sharing connection IS_A relationship. We also saw that red roses and yellow roses are both kinds of rose so we identify that as a semantic connection which again leads to different IS_A relationship.

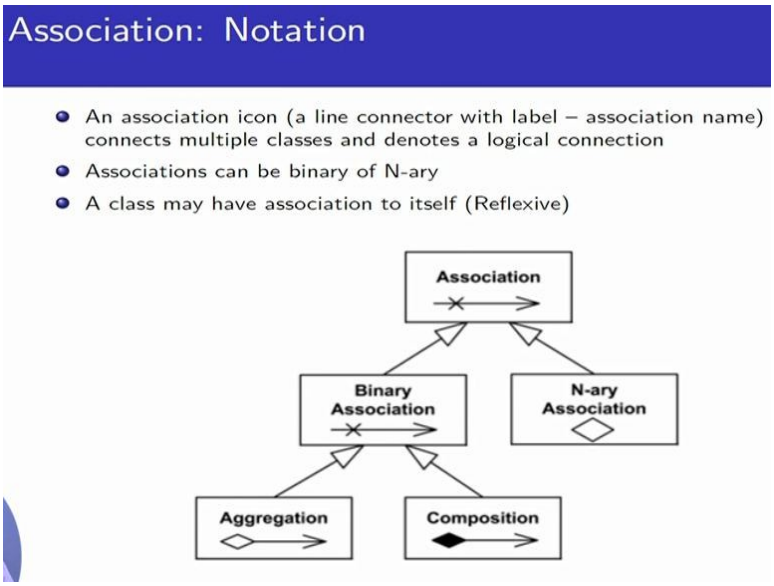
We saw petal is a part of both kinds of flower so this is a different semantic connection which gives rise to the part of or HAS_A relationship and we saw that there could be different other kinds of relationship also like symbiotic relationship as to how lady bug and flowers are related. Because possibly lady bugs protect the flowers from certain types of pest.

(Refer Slide Time: 02:40)



We have also seen that as the class, as the design go through different phases. Then initially we may just have an identification of an early relationship were we just kind of connect the two classes by a solid line and as we refine then may be able to identify what is the actual relationship in that and may qualify that association. This is another flower and petal initially just an association and refined to HAS_A. So we will see these representations take part in the class diagram.

(Refer Slide Time: 03:18)



So the first relationship which is quite important is association which says that the two classes are associated in a certain way. The association has a name and actually it is not two classes alone it could be between multiple classes and if more than two classes are involved then we say

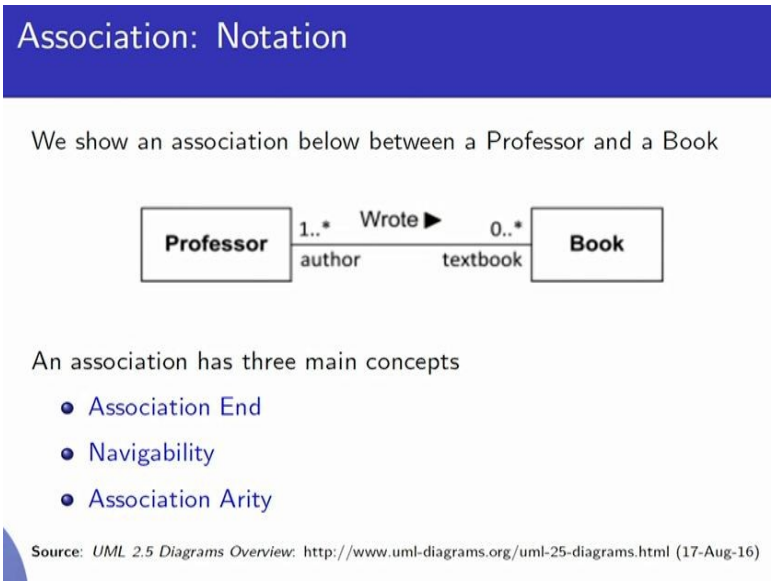
it is a n-ary or if three classes are involved we will say this is ternary relation/association if two classes are involved we say it is binary association.

So associations could be of multiple different arity but usually more often we will see that variety of binary association is what dominates the representation scenario. And interestingly a class may have an association with itself. So I have a class person I may have an association like this. So it says that person is associated with person which will mean that basically there is some relationship between two different instances of the person.

For example, we will see is married say this association means married to so on this side we have husband on this side we have wife. Both are instances of person but they are associated through the marriage. So this is a kind of a self-association or reflexive associations are also possible. And these are just different this is the kind of the UML diagram to specify the different kinds of association that we have.

So this is the root concept of association and then it could be binary as I said or it could be n-ary. Binary what is more frequent and binary association can be aggregation or composition. We have talked about these terms in earlier modules as well but certainly when we come across them in the class diagram we will again explain what these different binary association means. So this is a basic hierarchy of specialization in association itself.

(Refer Slide Time: 05:43)



So this is taking an example this how an association is expressed. So in association the simplest form is a binary one so one side there is a class professor other side there is a class book. So if I start the very being then I have professor, I have book, I join a line I say there is an association. But if I just say that the professor and the book are associated then only little information is communicated. So it comes a question of how the professor and the book are associated.

For example, it is possible that the professor is associated with book because professor reads book. It could be READS. Professor could be associated with the book because he has a book, he owns a book. Professor could be associated with the book because he authored a book he wrote a book. Professor could be associated with a book because the author of that book has mentioned the professor's name in the acknowledgement. So association could be of variety of different forms.

Even when they occur between two classes so we put a name to the association. Which try to semantically specify that what is being associated with what and how? The root is a name of the association and on the two sides we mean this is again optional. We mean see that when root associates professor with book then which role of the professor are we talking about. Because a professor could be an author, the professor could be an instructor.

The professor could be a mentor; the professor could be just acting as a faculty and so on. So here we are saying this association is referring to the author role of the professor. This association is referring to only the textbooks in the whole domain of books. So this is the basic structure of an association and whatever we write here at this two ends is specified in terms of the association end and we can have more information on this.

And the association end we could also have multiplicities which say how many instances of this class associates with how many instances of this class through this association. So what does it say what does it mean if we look at the multiplicities. On the professor side it is one dot dot any so it says that a professor is associated with the book one professor two professors, three professors any number of them which means that a book can have one or more professors as authors because they wrote the book.

This side we say that it is zero dot dot star that it could be zero book, one book, two books any number of books which say that a professor can write zero book it is possible that a professor has not written a book. So that is zero. He may have written one book. He may have written five books and so on. So in this way the different professor P1, P2, P3 different books B1, B2, B3 will have different kinds of actual relationships.

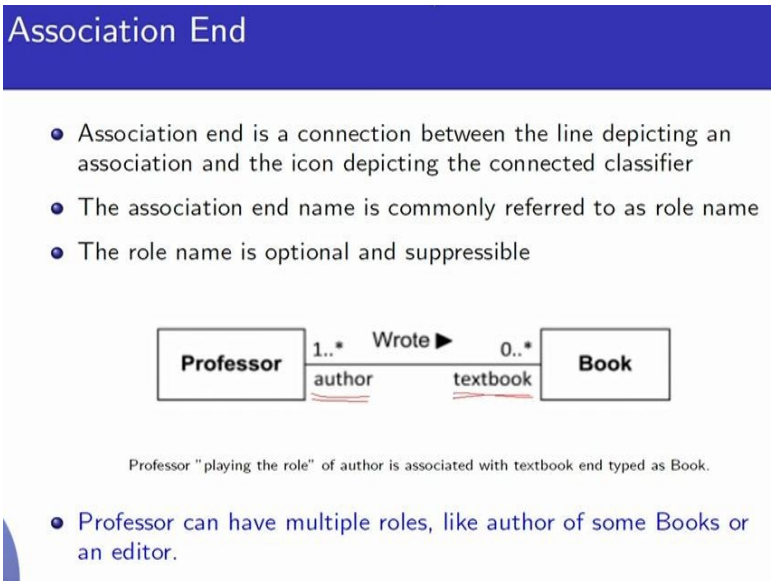
So it could be like this so it says that the book B1 is written by P1 and P2. Professor P1 has written two books B1 and B2. Professor P3 has written B3 alone. So this could be all these different kinds of when there could certainly be a professor P4 who has not written any book so not associated. But we cannot have a book B4 which is not associated. Why? Because a multiplicity here says it is one.

So if I have a book it must have a professor on this side which is understandable because you cannot have a book without an author. There must be one author at least. But you can have professors who have not written books. So these are the different key concepts that an association has and we will see what this mean. The association end we have started talking about it.

The navigability as to how you go from one end of the association to the other and the arity of the association. By the way when you write the name of the association you normally put an arrow, filled arrow which is called the reading order of the association that is professor wrote books so he will not read this association as book wrote professor this arrow says that you have to read it in this direction.

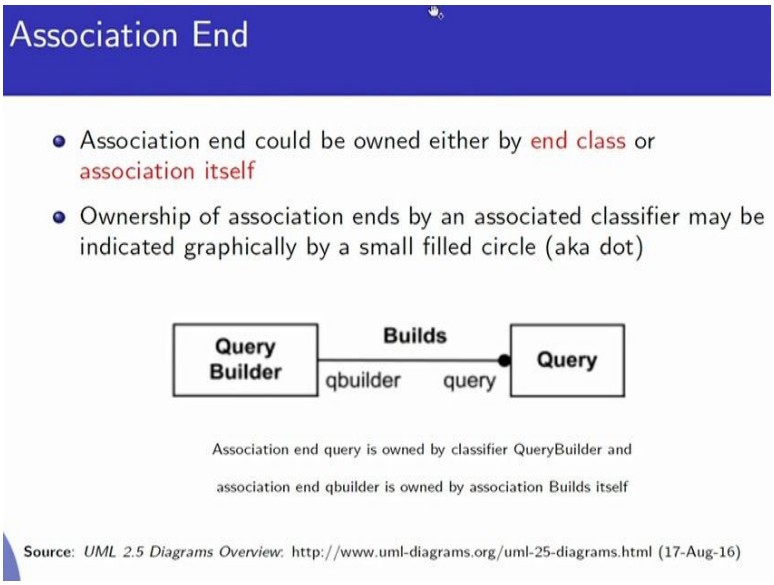
It is not again mandatory but it is usually convenient it will have the reading order also marked in the association.

(Refer Slide Time: 11:17)



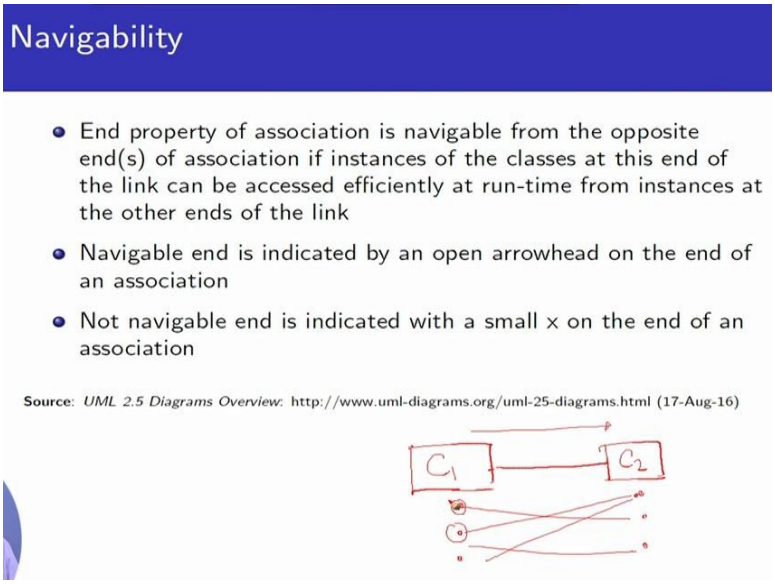
So this is a basic representation in the association end we have I have already talked about the role on the sides. The role of the professor, the category of books we have talked about. So this we have actually already discussed.

(Refer Slide Time: 11:38)



Now it is also possible the association end could be owned either by the end class or by the association itself. So that is what is represented by putting a dot at this end. So if we dot like this the name of the association is builds and so the association end query the name of this end is query is owned by the query builder. So query builder naturally owns that query. If not the builds association itself so association end could be specified in this manner.

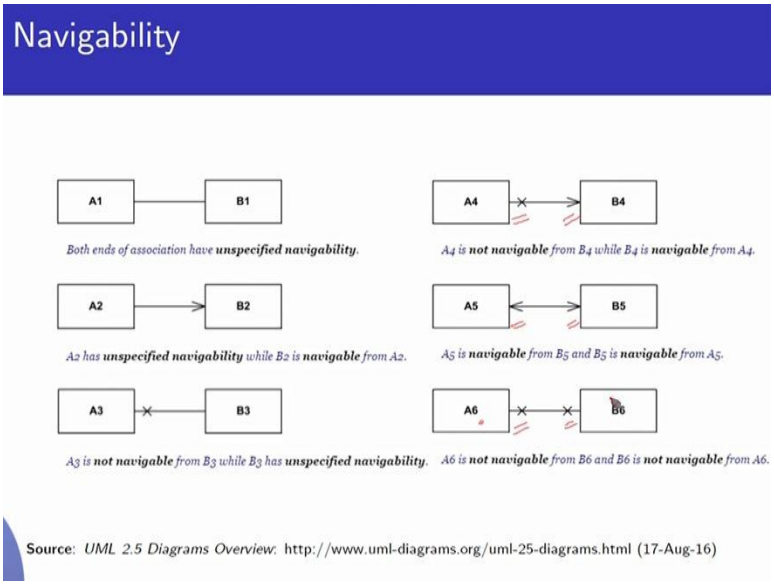
(Refer Slide Time: 12:24)



Let us see the question of navigability. That is if I have two classes associated then how can I go from one class to the other which means if I have objects of this class the association or the relationship is between this objects. Now is it that given an object here I can go to the other

related object. Is it that given the object here I can go to the related object on this other side and so on so forth?

So navigability tells us which direction or which directions in which I can go from one object on this side to another object in this side. So navigability can be defined in multiple different ways.
(Refer Slide Time: 13:28)



So these are the different symbols that are used. See major symbols one is if I do not specify anything is just a flat connection then we are saying that we do not want to specify the navigability so this is called unspecified navigability. I have not been able to model as yet as to how it can be navigated. Or I can have a directed arrow so it says that B2 is navigable from A2. So I can navigate them so given some A2 object I can find the associated B2 object.

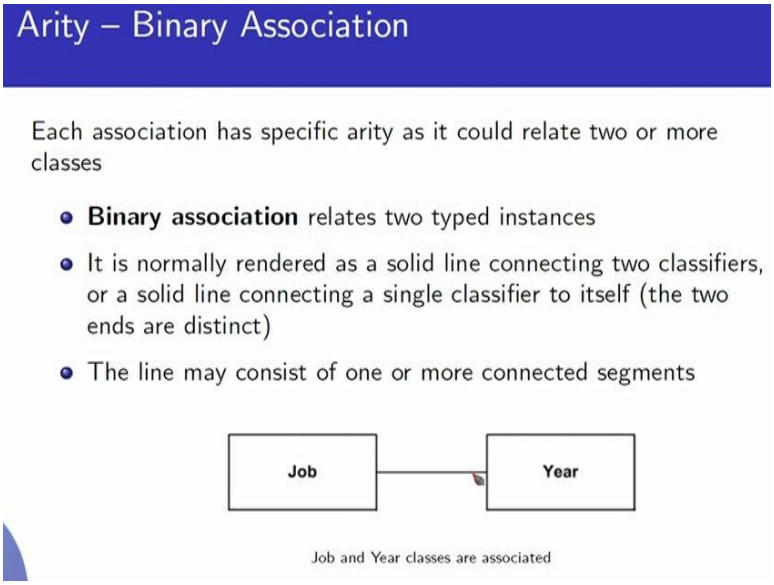
This is the basic idea. I could cross out navigation. If I cross out on one end, then it says that A3 is not navigable from B3. I know this is different from not drawing anything if we do not draw anything then it is unspecified that I may be able to navigate I may not be able to navigate it is not known but when you cross out it says that it is not navigable that is given B3 objects I cannot come through the A3 objects related A3 objects to this association.

So here if you look into this this side is not navigable and this side is unspecified. Similarly, more examples are given here. This is not navigable; this can be navigated. Here it is a

bidirectional navigation that you can have objects on any side and navigate to the other. This does not allow any navigation at all. So the object if you have an object here you cannot find the associated object through navigation of this particular association.

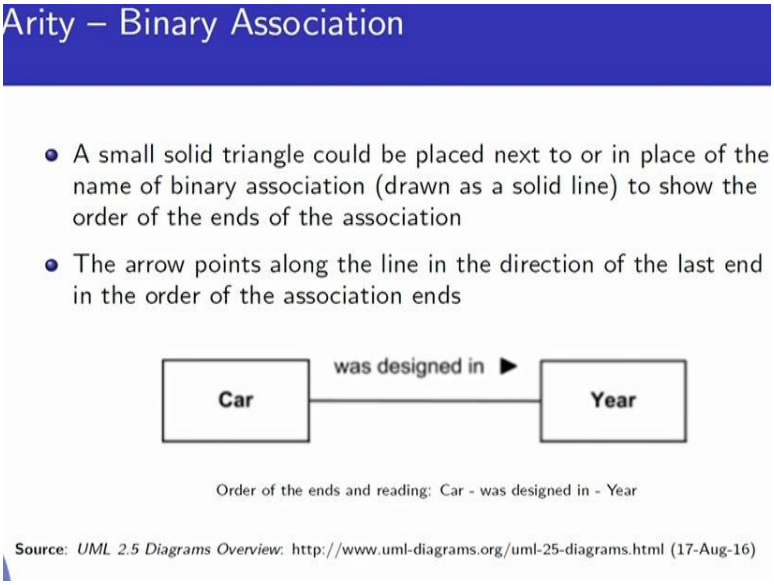
And the same is true from the B6 side coming back to the A6 side. So these are different navigability that the class diagram allows.

(Refer Slide Time: 15:22)



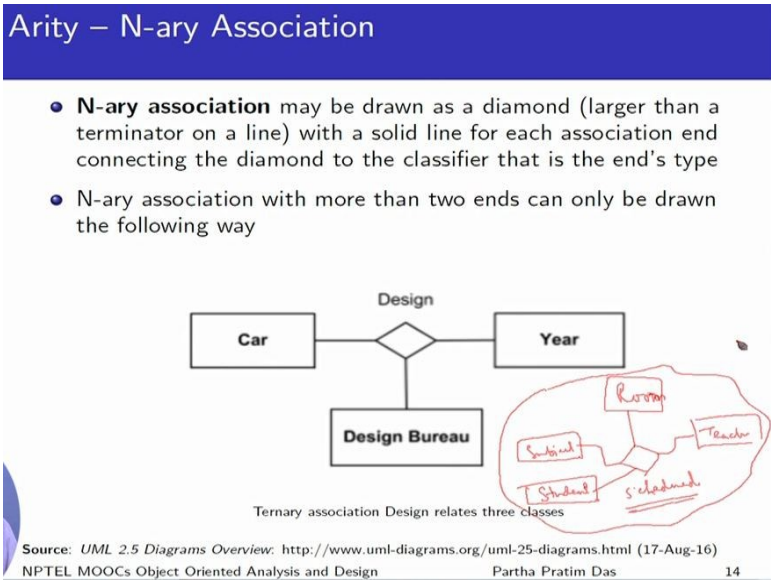
Next is the arity which is simple to understand. Most associations are binary and we have already seen how these are related to one line and along with that I could have all those navigability, reading directions, name and all of this.

(Refer Slide Time: 15:43)



So this example we have already seen. This is the name of the association and this is the reading direction and it is simply creating it binary association.

(Refer Slide Time: 15:58)



If I have an N-ary association then we use this kind of an open diamond, a big open diamond and connect the different classes on different sides. You give a name to this diamond which is the name of the association. Now certainly please do not think that since that since diamond has four corners you can use this only to connect four classes in an association. Association can have any arbitrary arity.

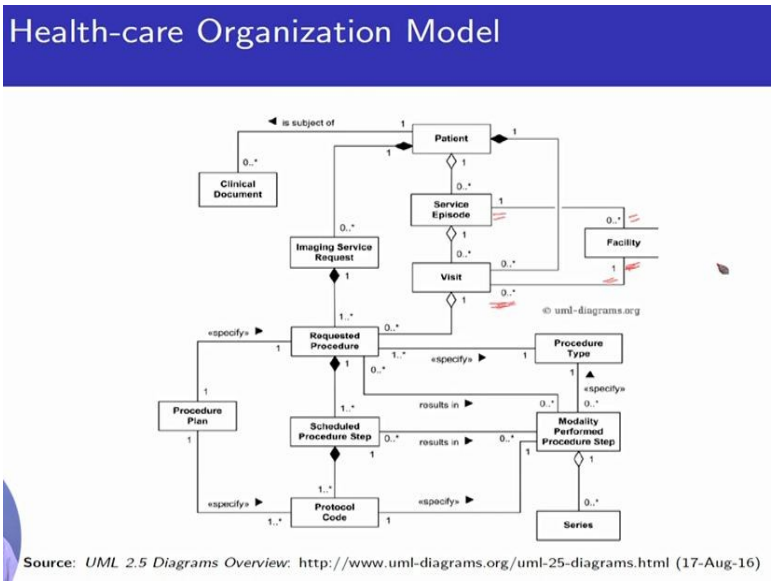
So it is possible that I can connect five, six, eight, ten different classes through an N-ary association. But of course it is true that as binary is the most common ternary is seen to quite a times. But association of higher arity is more difficult to understand more difficult to model and maintain so usually will not see them very frequently. For example, one nice example of say the ternary kind of association is suppose you are working on a timetable scheduling classes.

So you need to know what the room is. You need to know what the subject is. You need to know who the teacher is. So between these three classes the room so I have room, I have subject and I have teacher. So I can have a ternary relationship which is a say schedule that when is a particular subject scheduled in a room taught by a certain teacher. So this is a ternary relationship.

That will represent the information of the triplets having been related because a subject is held say in the second hour of every Tuesday and taken by certain teacher. Now along with this if we try to model the student's room to attend this course. Then I could put an attending presence in the relationship and this will become a relation with arity four. But certainly as you can understand this makes it more and more complex to analyze and follow.

So normally, we will try to decompose everything in terms binary relation and at times use ternary relation.

(Refer Slide Time: 18:54)



So this is just to show certain instances of different class diagram. So this is like a health care organization. Of course these examples have lot of details so I am just talking about some sampled aspects you should take time and try to understand these examples more thoroughly to understand what do they mean and will find you will soon be able to start understanding how to read this diagram and get the information from them.

For example, we can see this is a requested procedure possibly for certain ailment and we say that there is an association with modality performed procedure step. For that procedure something you need to be done. For example, if an ECG has to be done then every lead of the ECG must be added with certain liquid so that modality. So you are saying that there is a relation association binary association between this class and this class which results in.

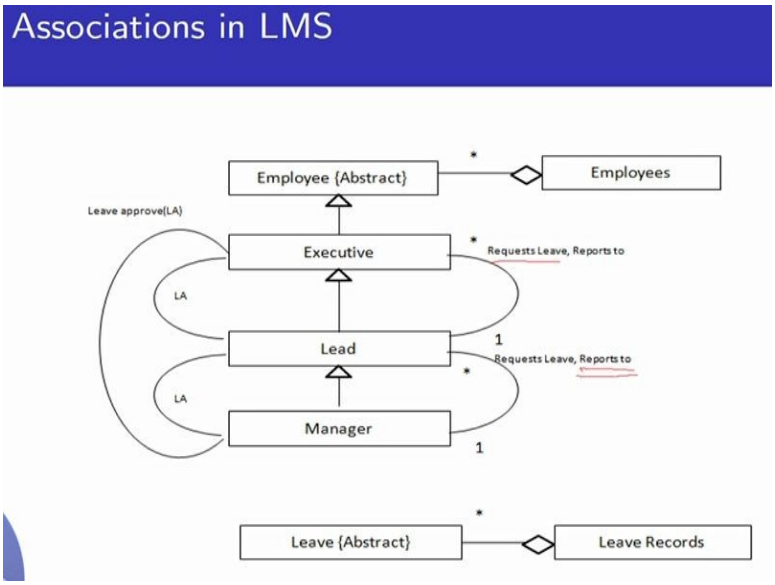
So requested procedure results in modality being performed in the procedure step and the multiplicity is zero dot dot star which means that a requested procedure may not result in to any of this a modality performed procedure step may not have raise from the requested procedure. But it could have any arbitrary number of associations as well. So you could see in this way that different kinds of binary association are involved here.

So I would just request and in this the navigability is not specified. You do not see any directions at the end of the arrows so it basically says that the navigability is unspecified. So for example

the facility and visit have an association between them so a facility may be visited by server but one visit will always be a unique facility. You cannot at a time be at two facilities. So this side is one whereas this side is zero dot dot star.

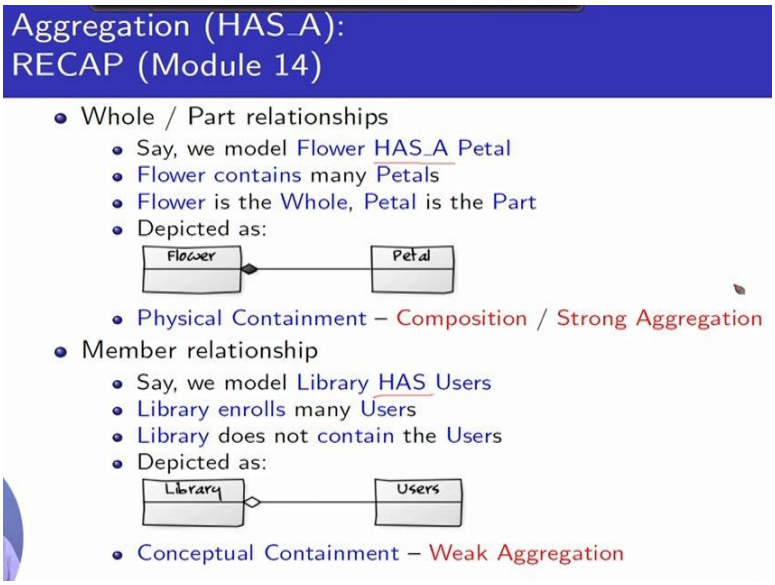
So please spend some time and I have also given the url pointer is to where I have taken this diagram from. If you go to this particular url which is the UML-2.5 diagrams overview discussion in that portal. You will find also the detailed description or more description of what this problem are and how this class diagram should behave.

(Refer Slide Time: 21:54)



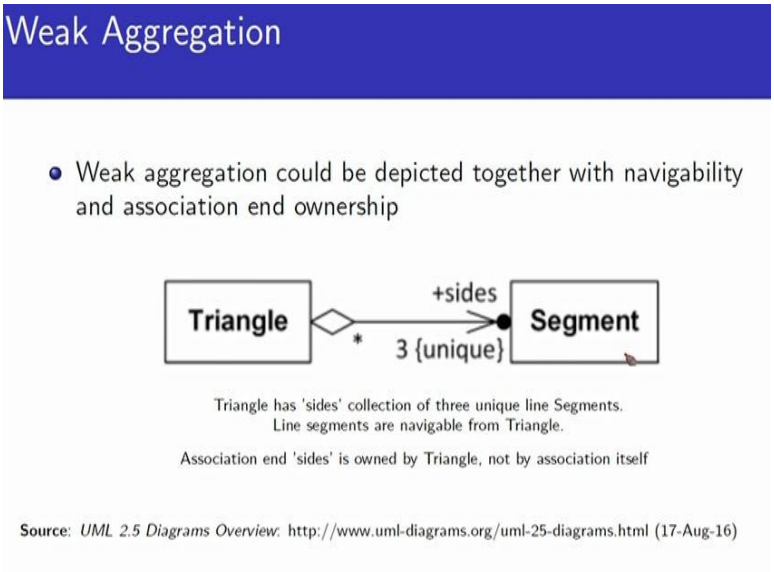
These are different associations in the LMS system particularly relating to request from reporting relationships and so on. So these are different binary associations.

(Refer Slide Time: 22:08)



The next is the aggregation which we know HAS_A relationship and we have seen earlier that there is a difference between HAS_A and HAS which is a membered relationship. So flower HAS_A petal because petal is a part and parcel of flower but library HAS users because library is associated with the users but users are not component part of the library.

(Refer Slide Time: 22:42)



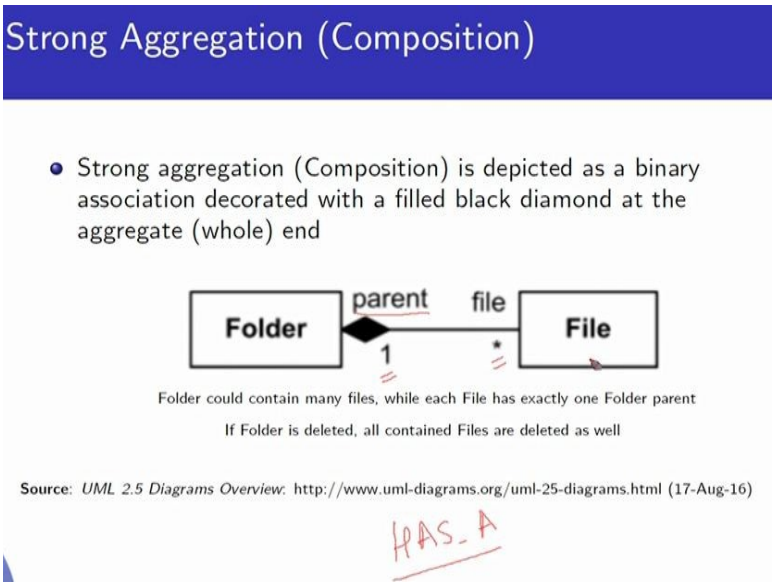
So this association needs to be represented. So based on this HAS_A and HAS there are two kinds of association which are specified. There are known as aggregation association. The first time is a weak aggregation. Weak aggregation is represented by an open diamond and has two classes again. This is a binary aggregation and this is a HAS or membership kind of aggregation so the triangle has sides or line segments.

Now it is possible that you have a diagram like this you have a diagram where you have so segment may be associated with multiple triangles. So when you look at from the segment side you put this aggregation to be at star which says that a segment could be by itself also not a part of the triangle. So we say that a segment could be associated with any number of triangles. But a triangle will be associated with three segments.

And then you give it a name and you say that this segments are called sides but the end you put that name. so whenever you have that situation between library and user and this kind you say this is a weak aggregation and along with that we can noted further information so now we are saying the same relationship of weak aggregation this says there has to be three and now we are saying that this is unique.

Because certainly if I have one segment here one segment here and one segment here this is not a triangle all three segments do not give a triangle they will have to be unique non-overlapping segments so that is what is being said here. We have also specified the navigability that from the triangle you can go to the segment. From the triangle you can decide on that which is quite understand.

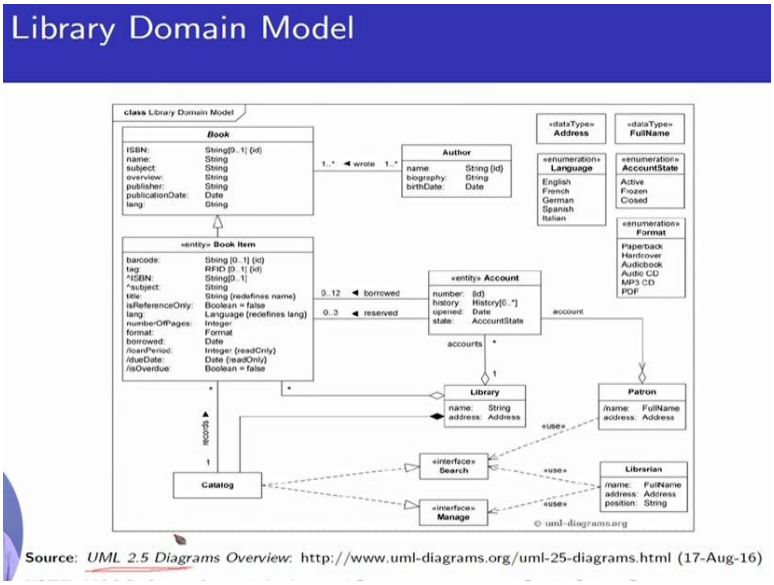
(Refer Slide Time: 25:03)



The other kind of aggregation is a composition where one is a part of the other so if I have again go back to the flower and petal then we will have something like this and this is designated by a filled up diamond. So if the diamond is filled up we say it is strong aggregation so you are saying between the folder and the file there is a strong aggregation because folders contain files so if you delete the folder files will also get deleted.

So they are a part of the file. So if you destroy a flower the petals will also be destroyed. So whenever you have that so this actually is model HAS_A or part of relationship in terms of a binary association. Certainly you can put some name at this end and so on. So the multiplicity is clear because the folder can contain arbitrary number of files but a file can remind only in one folder.

(Refer Slide Time: 26:22)



So now if we go back and take a further look again in to the library domain model which we had seen in the last module as well we can see for example you have accounts of users and we have library and there is weak association between them. This is a weak association between them and it says that the account will belong to the specific library but a library may have multiple such accounts where the users are coming in.

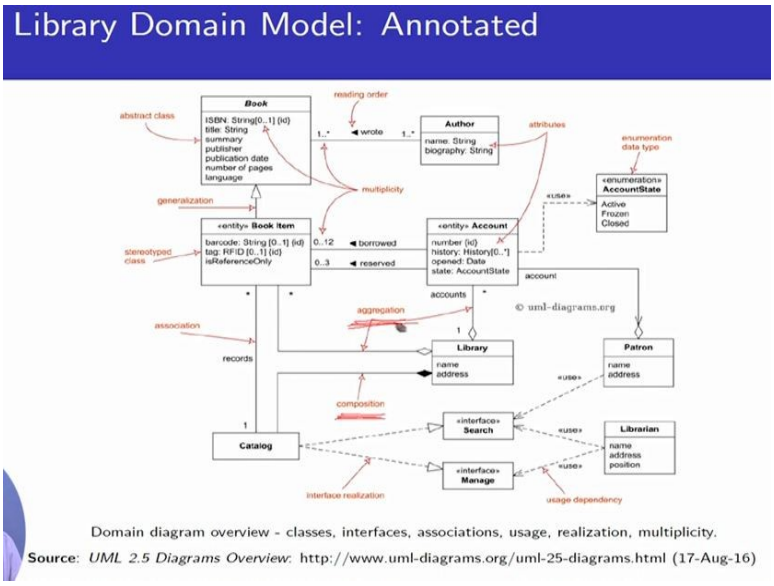
Similarly, it says that catalog which list the different books and the library are associated in a strong aggregation and nothing is specified which means that this is one-one that a library has a

unique catalog and a catalog belongs to library. Interestingly here for example the designer has decided to model the association between library and book item as a weak aggregation not as a strong aggregation. Why?

Because the books are that a library is just in the collection but it is not like if you remove a book or you add 10 books the state of the library changes but basically a library means a library but if you remove a catalog then the library will not be there. So this is being represented here in terms of a weak aggregation. We have multiples here which tell you that a library can have more any number of copies of a particular book.

There is no multiplicity given here it says it is one certainly a specific copy of a book can exist in one library only. Similarly, you have other associations here which are records, the catalog, and books. One catalog, records any number of books. You can see other different aggregation relationships also at other places and try to understand the diagram better. More specifications are given in the site. And you could check out those.

(Refer Slide Time: 28:56)



And this is again we also had seen this diagram, this is just same diagram but it is only that this annotated now. So if you wanted practice then you can just make reference to this. This is a composition a strong aggregation. This is an aggregation or a weak aggregation and so on.

(Refer Slide Time: 29:25)

Summarize we have several different kinds of association relationship between the classes in a class diagram. And these associations could be represented in very simple link terms or they could be qualified. They could specify multiplicity have name, have reading direction, navigability and so on. And the arity could vary from being two to being any arbitrary number, the N-ary association.

And specifically we have taken a loop in to an aggregation kind of relationship where there are two kinds the weak aggregation and the strong aggregation. Weak aggregation represents HAS relationship and strong aggregation represents HAS_A as a part of relationship. With this we will conclude this module and the next module we will continue in discussing the remaining kinds of relationships that classes may have.