Object-Oriented System Development Using UML, Java and Patterns

Professor Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture 29
Introduction To OOAD

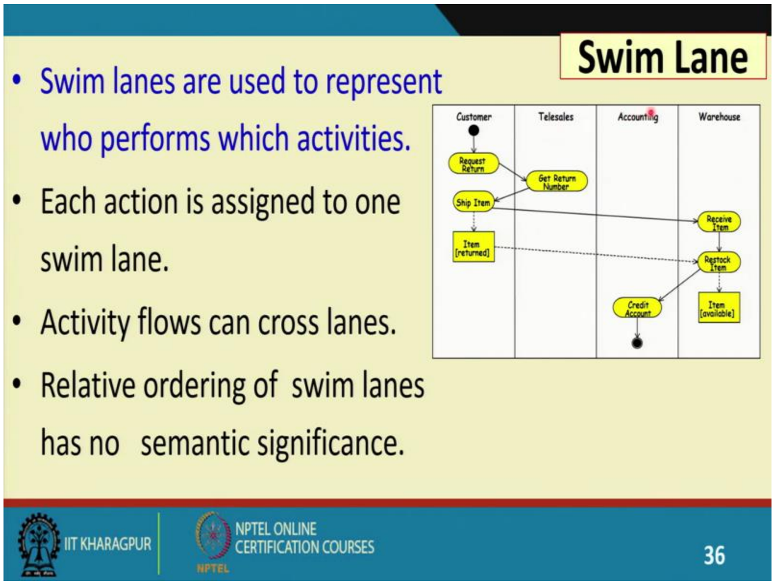
Welcome to this session.

In the last session, we were discussing about the activity diagram, which is an important UML diagram. Activity diagram is extremely powerful. Though, has vague similarity with flow chart but we can model things which we cannot model using a flowchart. For example, who performs a certain activity, the object flows, parallel activities, synchronization and so on we cannot model using a flow chart. The flow chart does not have any way to represent those aspects, but in the activity diagram, we can represent this and therefore the complicated business processes we can easily represent using an activity diagram, which become very easily understandable.

Just to give an example, let us say the admission to an academic institute like IIT is extremely complex. There are tests, there are various levels of the test, various entry points, undergraduate, postgraduate, and then the student reports the verification. The student is assigned hostels, the student pays fees, issued identity card, library card and so on. If we write a description of the student admission process in academic institute, various types of students, it may span large number of pages. But we can use an activity diagram to represent very accurately the steps that occur and who carries out that step. We had discussed about some of the constructs used in the activity diagram. Let us continue with what we are discussing.

In the last session towards the end, we are discussing about the swim lanes. The swim lanes can represent who performs which activities.

(Refer Slide Time: 2:41)

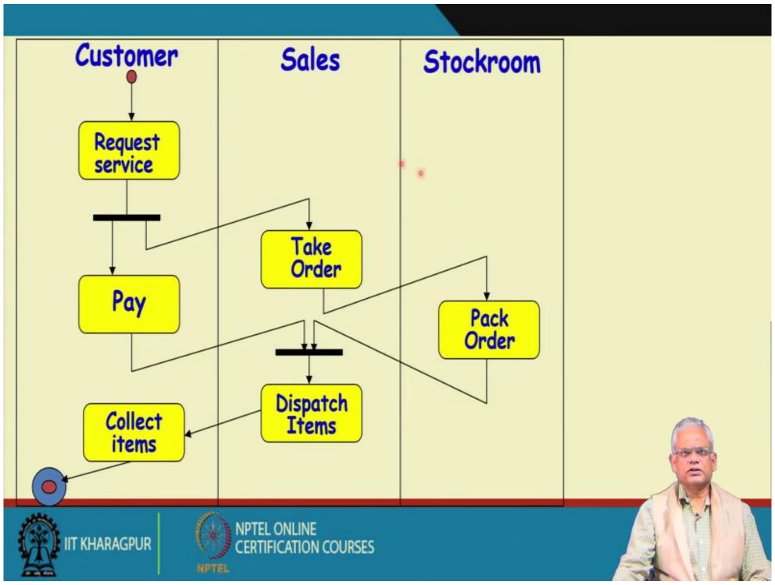


Swim Lane

- Swim lanes are used to represent who performs which activities.
- Each action is assigned to one swim lane.
- Activity flows can cross lanes.
- Relative ordering of swim lanes has no semantic significance.

Now, let’s look at the one that we are discussing. These are the different swim lanes here. And each swim lanes have a responsible component who carries out these may be departments or maybe some set of objects and so on. In the swim lanes, we represent the activities that are assigned to that component, just to give an example of how the activities are represented in swim lanes, just look at this example (in the below slide).

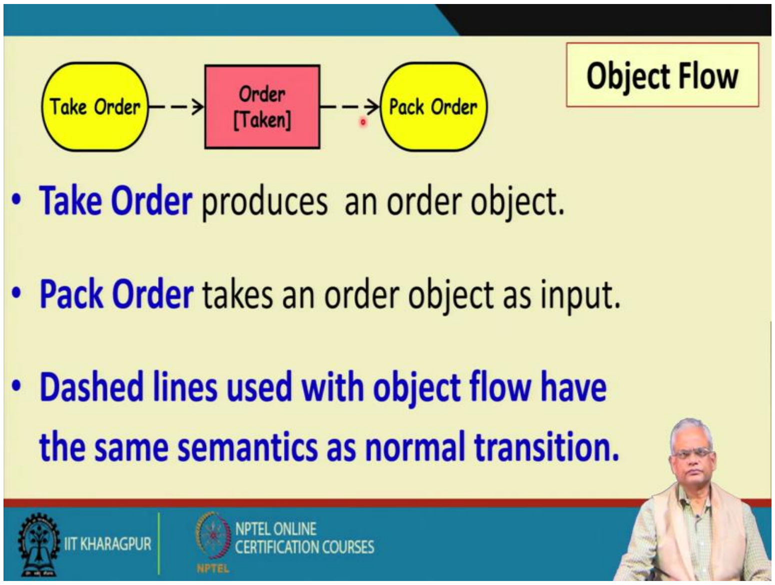
(Refer Slide Time: 3:25)



See here, there is a customer, the sales team and the stockroom. The customer requests some service. The salesperson takes the order and at the same time wants the service is requested, the order is taken by the sales team and also the customer pays. And once the payment is over and the taken order is used to pack the order and both payment and order packing is over then the item is dispatched and the customer collect the item. Maybe in a food counter, you go there place the order and there is somebody taking the order, the sales team and the customer is paying up.

And then the order is prepared and packed, and then once the payment receipt is submitted and the packed order arrives then the item is given out in a cover and the customer collects the item and note here that we have successfully represented not only the activities flows, but also who is performing the activity by using swim lanes.

(Refer Slide Time: 5:16)



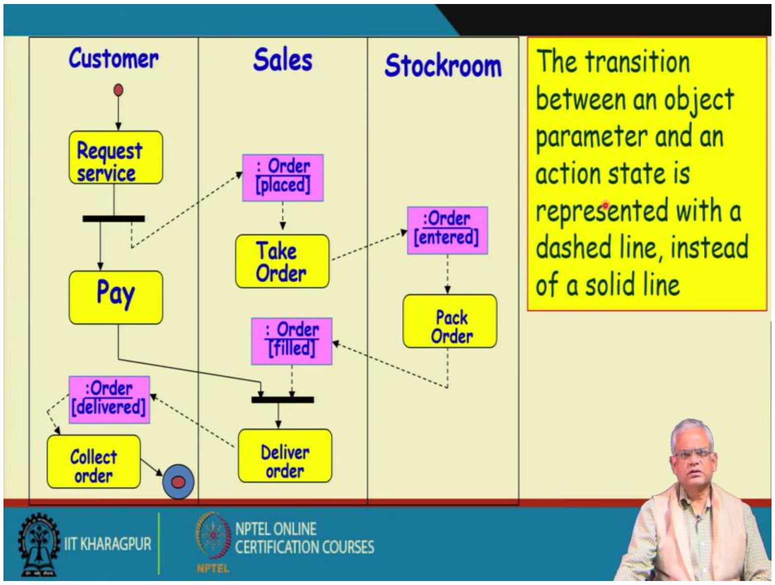
Object Flow

- Take Order produces an order object.
- Pack Order takes an order object as input.
- Dashed lines used with object flow have the same semantics as normal transition.

A very powerful mechanism in the activity diagram are the object flows. An object is represented using a rectangle here (as shown in the above slide). Name of the object here is Order, and we had also representing the state of the order if we want. Within brackets represent state of the object. Here [Taken] represents that the order has been taken, the order is in the taken state and the object flows between two activities are represented using dashed lines. The control flow is represented using solid lines, but object flows are represented using dash lines. Many case tools support activity diagram. You can see how you can draw activity diagram. The implication of this diagram is that the Take Order activity produces an object, which is the order, and the state of the object is the order has been taken and then the taken order is input to the Pack Order activity.

And as we are mentioning, please remember that the object flows are represented using dashed lines between the activity and the object and also between the object and the activity.

(Refer Slide Time: 6:59)



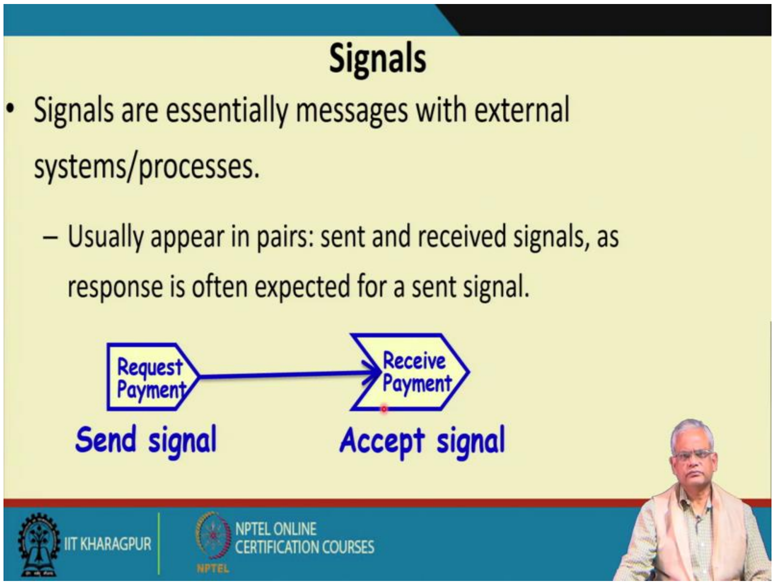
The transition between an object parameter and an action state is represented with a dashed line, instead of a solid line

This is the same example with the object flows (in the above slide). Here the order object possibly the customer writes down the order and that's the order object and it is taken by the sales. This requests service creates an object here and the order state is placed and then using that sales team takes the order and then enters the order.

So, the state of the order changes to entered and then that order is entered order goes to the pack order, which is done in the stockroom. And see here that the Order object flows occur, the order object flows are represented using dashed lines. Whereas the activity flows are represented using solid lines. And once the order has been packed, then the filled order is taken place in Sales lane. In the synchronization bar, where both the filled order and the payment slip is given by the customer is join.

And then the sales deliver the order and then the delivered order is collected by the customer and that is the final node or the end activity node. The important thing to note is that the traditional control flows represented using a solid line, where is the object flows, are represented using a dashed line.

(Refer Slide Time: 9:18)



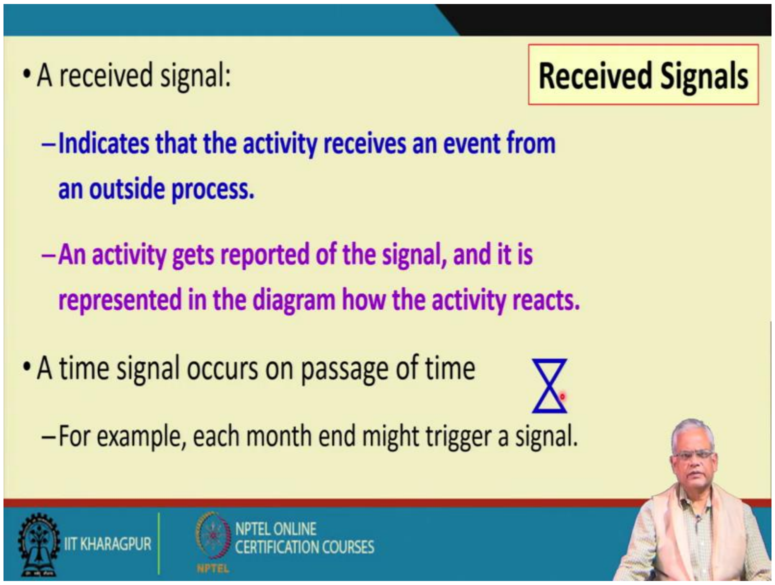
Signals

- Signals are essentially messages with external systems/processes.
  - Usually appear in pairs: sent and received signals, as response is often expected for a sent signal.

Another feature of the activity diagram are the signals or events. This is a signal or event (as shown in the above slide) which is given to another activity. For example, request payment event may be given to the customer which may be a missed call, which may be an email or a SMS. This is a signal and then the customer makes the payment and the system waits until the payment arrives. And that is represented using the accept signal.

So, there are two things in the signal handling. One is the send signal symbol and the accept signal symbol and typically they occur in pairs and the system waits once the event is sent, until the corresponding recipe event occurs the system waits.

(Refer Slide Time: 10:46)

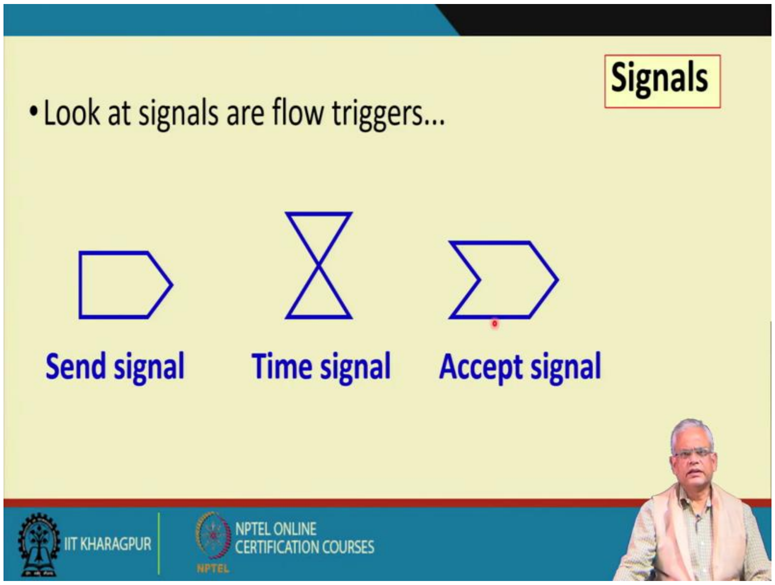


Received Signals

- A received signal:
  - Indicates that the activity receives an event from an outside process.
  - An activity gets reported of the signal, and it is represented in the diagram how the activity reacts.
- A time signal occurs on passage of time ⧗
  - For example, each month end might trigger a signal.

Typically, the signals help to communicate with outside parties who are a different system located elsewhere. They have their different process and this is a powerful mechanism where one process in some system collaborates with another system. So, this outside process, the event is sent and the response is received, and once the response is received, we also represent how the waiting activity reacts to it. One important use of the signals is a time activity. For example, at the end of every month, the salary needs to be disbursed. We can represent that this is the time signal which gets generated at the end of every month. And we might have a salary slip generation and payment activity, which is waiting for the signal of the end of the month and once this generates the signal. The time signal once this is generated then the corresponding action, which is waiting about salary slip disbursal and payment credit activity will take place.

(Refer Slide Time: 12:40)

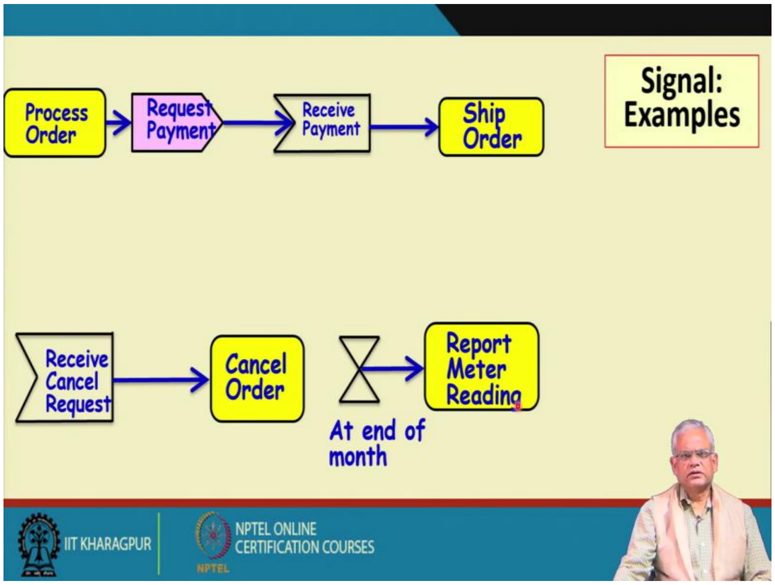


Signals

- Look at signals are flow triggers...

Just to summarize the different signals we will observe the above slide. Send signal represented with the symbol shown in the above slide, the time signal typically generated using a timer or a clock and then the received signal. The activity waits until the signal it is waiting for, it is accepted or is arrives at the accept signal and then the activity gets triggered.

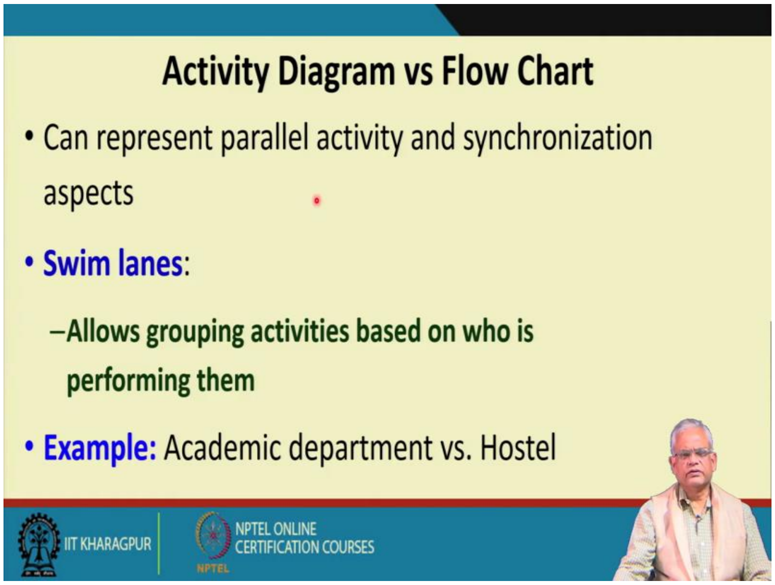
(Refer Slide Time: 13:16)



Signal: Examples

This is just one example (in the above slide) of signal is that after the Process Order has been processed, request payment signal is sent to the customer. And once the customer makes the payment, the Ship Order activity gets triggered otherwise, as long as the payment does not arrive, it just keeps and waiting here in Receive Payment. As you can see, these signal concepts make activity diagram very powerful modelling technique. Another example that the customer can at any time cancel the order becomes very difficult to model without using signals because the customer can cancel the order any time after the order has been placed. We can easily represent that using a signal activity. This is the signal (as shown in the above slide), received signal for the cancel request and the cancel order activity is triggered only when the signal is received. If the signal is never received, the order does not get cancelled and the order is dispatched. This is a time signal (as shown in the above slide) at the end of every month, the meter reading is reported to some other activity or maybe in the end of every month a salary, disbursed salary and so on.

(Refer Slide Time: 14:56)



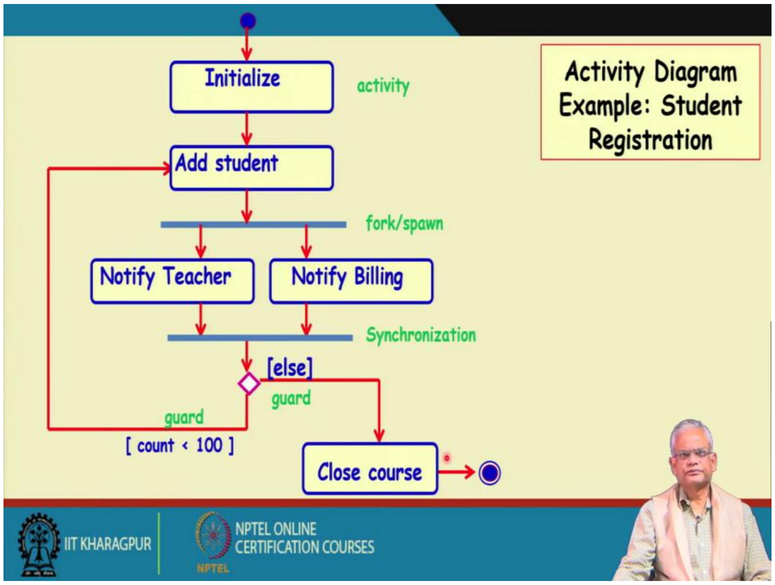
Activity Diagram vs Flow Chart

- Can represent parallel activity and synchronization aspects
- **Swim lanes**:
  - –Allows grouping activities based on who is performing them
- **Example:** Academic department vs. Hostel

We just saw that the activity model is a much more powerful modelling technique than a simple flow chart even though there are vague similarity of a control flow. But then the activity diagram has many constructs which make it much more expressive and powerful compared to a flowchart. For example, the parallel activity, synchronization, swim lanes, which allow activities to be assigned to specific entity which is performing.

Just an example that whether the academic department should carry out an activity or hostel should carry out an activity, the object flows and so on. A many feature you can just think over again, that how do the flowcharts and activity diagrams compare. What are the aspects of a business process that will be handicapped or hard to model without an activity diagram and you are required to use a flowchart.

(Refer Slide Time: 16:16)



Activity Diagram Example: Student Registration

Initialize

activity

Add student

fork/spawn

Notify Teacher

Notify Billing

Synchronization

[else]

guard

guard

[count < 100]

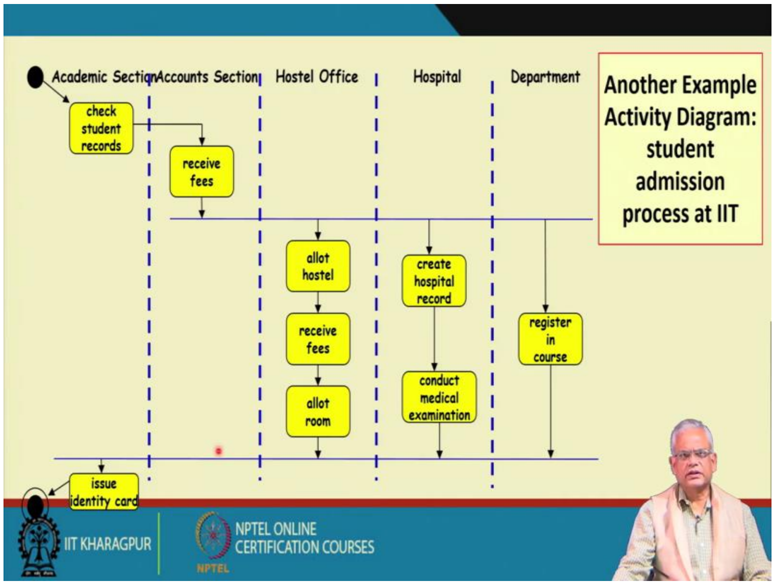
Close course

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

This is an example of a student registration process. This is the initialize activity is represented using a rounded rectangle. And then this student registration in a course. The student initializes where the student registration is zero number of student registration or more numbers. Let say one student registers we add the student. The teacher is notified of the student who is registered, and also the billing system is notified to build the student for the course. These two activities occur parallelly and as soon as these both these activities complete that the synchronization bar, then if the students count < 100, then we can add more student. If the student count =100, either we can write else or count equal to hundred. Either ways the activity is closed and these student registration use case.

(Refer Slide Time: 17:39)



Another Example Activity Diagram: student admission process at IIT

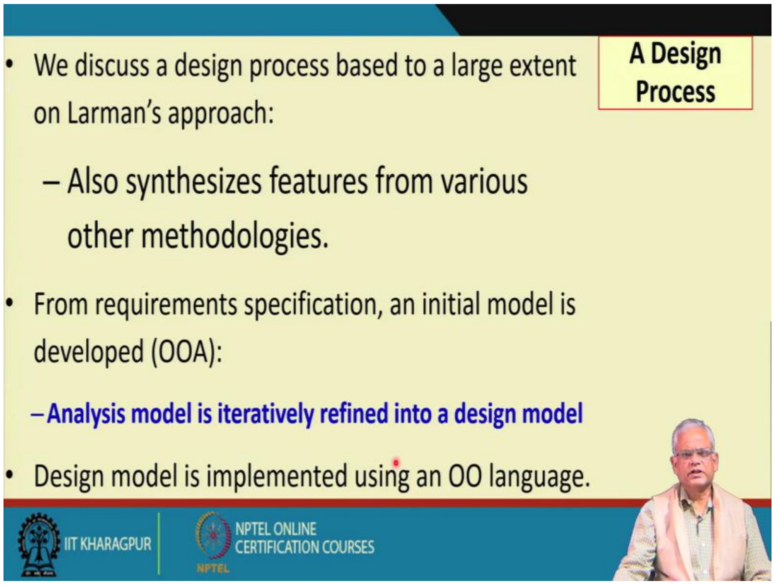
This is another example of an activity diagram of the student admission process at IIT (in the above slide). Here there are different departments which take place in this admission process: the academic section, the account section, the hostel office, the hospital the department etc.

The student initially reports to the academic section who check the records of the student and then the account section receives the fees. The hostel office, Hospital and for Department there is a parallel activity starting. They can occur in any order. First, maybe the allot hostel, the student may go and get the hostel allotted, or he may go to the hospital and create the hospital record or may visit the department and register in the courses.

The hostel office allotted hostel to a student. The specific hostel fees are paid and the room is allotted and the hospital conducts medical examination. And after all these activities are complete, then the identity card is issued in the academic section and the admission process ends.

So, far we have been looking at the UML modelling techniques. Those were necessary to start our design process an important objective of this course. Just knowing the UML construct, you cannot really design. We can use those to document and help in the design process, but we must have a design process at the step by step activities that you will carry out to able to design and in that process, we will use the various UML diagrams. And after the initial design is complete, we will use the design patterns to arrive at refined designs, and even during a design process, we might use the ideas of the design patterns to arrive at good designs. So, far we have looked at the UML and the different construct of UML and from now on, in this course, we will discuss about the design process and we will subsequently discuss about the design patterns. How to use the design patterns to arrive at a refined and improved designs.

(Refer Slide Time: 20:47)



A Design Process

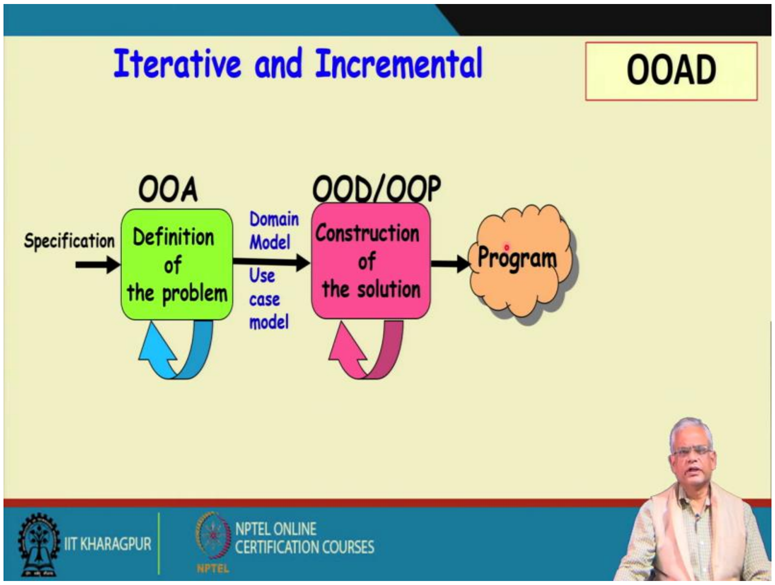
- We discuss a design process based to a large extent on Larman's approach:
  - Also synthesizes features from various other methodologies.
- From requirements specification, an initial model is developed (OOA):
  - Analysis model is iteratively refined into a design model
- Design model is implemented using an OO language.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

Now, let's look at the object-oriented analysis and design typically called OOAD. The design process that we discussed here is largely based on the Larmans approach, but also, we use ideas taken from various other methodologies. From the requirements specification, we start and we develop a graphical initial model. And that we call as the analysis process. And once we got the analysis model, we refined it into the actual design model and once we have the design model much of the code can be easily or mechanically written from the design.

And we were using a case tool, much of the code gets automatically generated by the case tool. And once we have the design model, we either generate the code using a case tool or we mechanically write the code using some object-oriented language.

(Refer Slide Time: 22:08)



Iterative and Incremental

OOAD

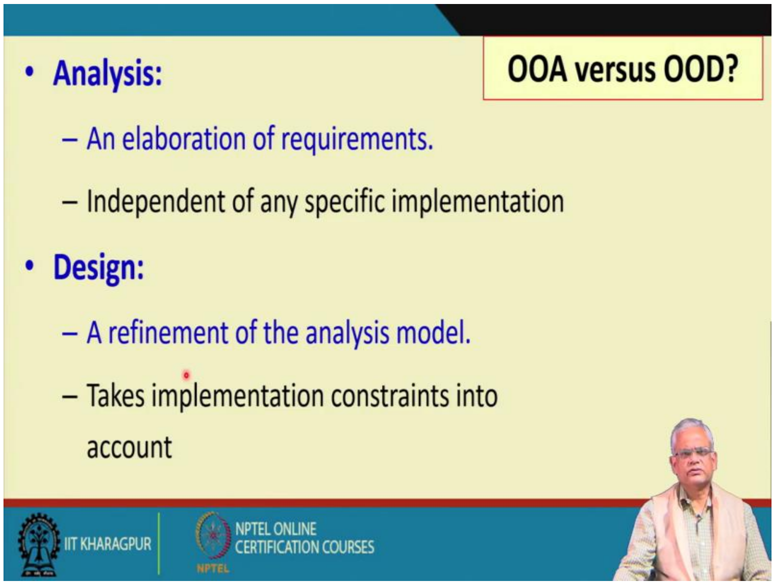
OOA
OOD/OOP
Specification
Definition of the problem
Domain Model
Use case model
Construction of the solution
Program
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

This is the representation of our design process (in the above slide), which is both iterative and incremental. Initially, we do the object-oriented analysis. In the object-oriented analysis (OOA) part as you can see, we use the specification or maybe some text documentation on maybe the requirement specification that has been worked out in the form of an SRS document. We use the SRS document and through a set of steps, we carry out and get the outcome here of the analysis process into two documents. One is called the domain model and the other is called use case model.

These are two important output of the analysis activity and the analysis activities iterative we start with an initial use case model, we refine it and then we get a domain model, we keep on refining it. And finally, we have these two documents ready at the end of the analysis step: Use case model and Domain model. And then we have the object-oriented design and programming activity that starts. The analysis activity focus on the definition of the problem. In analysis activity, just understand the problem, as has been described by the customer and then represent them using suitable diagrams.

On the other hand, in the design, we transform the analysis model to a model which can be easily translated to good quality code. We will take several examples where we will initially perform the analysis activity and we will get the domain model and the use case model and use this to get the class model and so on. And finally, much of the program can be automatically generated using a case tool or we might manually write the program almost mechanically.

(Refer Slide Time: 24:48)



OOA versus OOD?

- **Analysis:**
  - An elaboration of requirements.
  - Independent of any specific implementation
- **Design:**
  - A refinement of the analysis model.
  - Takes implementation constraints into account

But an important question here is how does object-oriented analysis and object-oriented design differ from each other? We have given some hints in the previous discussions we had over the last few slides discussing about the analysis activity and design activity. If you noticed that we were mentioning that in the analysis activity, we concentrate on the requirements it has been given to us. We understand the requirement, elaborate and develop some models of the requirement. We can say that using the analysis activity, we create a model of the requirement. We don’t consider how it will be implemented and so on. On the other hand, in the design process, we refine the analysis model, we take the analysis model and do some activities. All these will become clear in next couple of sessions where we take some examples and we work through the analysis steps and then the design steps.

We refine the analysis model into a design model, which is easily translatable to the code. We can therefore say that the design model is the model of the solution or the code. Where is the analysis model is the model of the problem. The design model is the solution of the analysis model. Here we take into account will it run concurrently, will it use a database, Will it back up, Will it to authentication, and so on.

With this basic understanding of the analysis and design activity, we are now ready to look deeper into how these steps are carried out because object-oriented analysis also consists of several steps which are carried out iteratively, and the design process also consists of several steps which are carried out iteratively. We will look at that in the next session. But before that let me just mention one more thing that in a procedural design the analysis and design are typically carried out very differently using different notations. For example, the analysis may occur through structured analysis techniques. They have different notations and different techniques. The design activity is carried out using structure design, which has very different notations. But in UML we will see that there is no hard and fast distinction between the kind of models or kind of the diagrams that are useful during analysis and not useful during design and so on. The same class diagram and so on will be used both in the analysis and design.

So, the distinction between analysis and design is not very marked. We can observe them if we are careful, otherwise they will appear like seamlessly transitioning from analysis to design activities. We will look at the design, the details of the analysis and design activities in the next session.

We are almost at the end of this session, we stop here.

Thank you.