Software Conceptual Design
Dr. Sridhar Iyer
Dr. Prajish Prasad
Dr. T. G. Lakshmi
Indian Institute of Technology, Bombay

Lecture - 24
Abstractions in Modelling

In the previous video, we saw how the FBS framework mapped to the software development process.

Yes, in week 3 we saw another set of diagrams, the unified modelling language diagrams. The process of representing the solution designed using representations like FBS and UML is referred to as software modelling.

Modelling is a way of creating an external, explicit representation of the system to be built based on the requirements we create an FBS representation to expand an idea or concept based on the system which we want to build.

So, once we have this representation, we can go into some more detail and create UML diagrams for the software system to be built.

So, one of the key differences between these modelling representation is a level of detail in them, right?

Yes, you are right. So, these levels of detail are known as levels of abstraction.

(Refer Slide Time: 01:19)

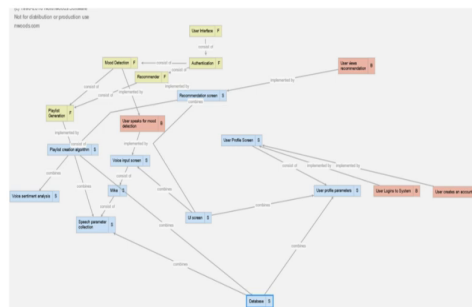Abstraction is a way of hiding details in the design at each stage of the design process.
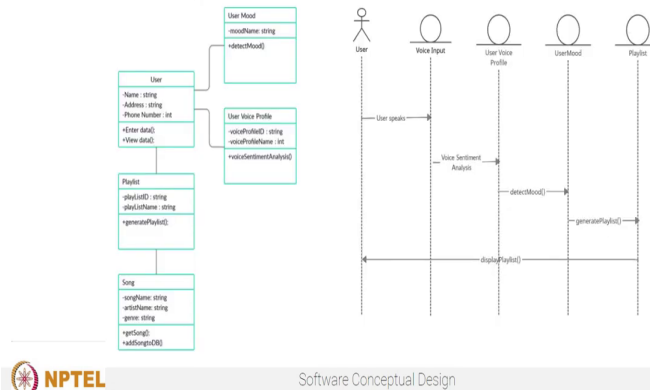
(Refer Slide Time: 01:33)

In week 2, we create a phase graph which expands on the function structure in behaviour of the design solution, it is an example of a high level extraction of the system.

(Refer Slide Time: 01:46)

We then add more and more details to the solution design. In week 3 we were introduced to the UML diagrams, they correspond to the medium level abstraction of the system.

(Refer Slide Time: 02:00)

In think and link we saw how we can expand from the FBS model to the UML model. The functions in the FBS graph can be expanded to create use cases, similarly, the class diagrams can be detailed from the structures. The sequence diagram actually describe the different behaviours in the system.
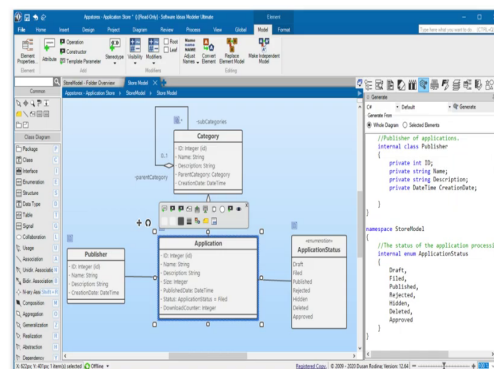
(Refer Slide Time: 02:30)

Then these UML diagrams can be further detailed out to create models such that there is a close correspondence with the implementation code. For example, every class in a model could map on to one or more classes in the implementation code.

(Refer Slide Time: 02:56)

There are also various tools which automatically generate code from such detailed UML diagrams. So, what we are doing is that we are adding more and more details as we progress across the various levels of abstraction.

Yes, this is an important takeaway. One way to create a software design is to continue adding implementation details as we progress.

Ok. So, another point is that these modelling techniques they are not the same right? So, do they serve different purposes?

(Refer Slide Time: 03:43)

That is a good question. Learners, let us reflect on this question what are the different purposes for which these modelling techniques can be used? Pause the video write your answer in the notebook, and then resume.
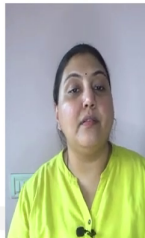
For example, the FBS graph is a vehicle for communication and idea generation when we want to discuss key ideas of the software with other members of team a modelling technique like FBS can be used.

These ideas are detailed out using modelling representations like the UML models these UML models can be used for analysis and comprehension, like what we saw in VeriSIM.

In the latest example, we saw that when we generate UML diagrams with greater level of detail, the automatically code gets generated. These detailed UML diagrams can be given to developers who can start implementation in a specific programming language based on these models.