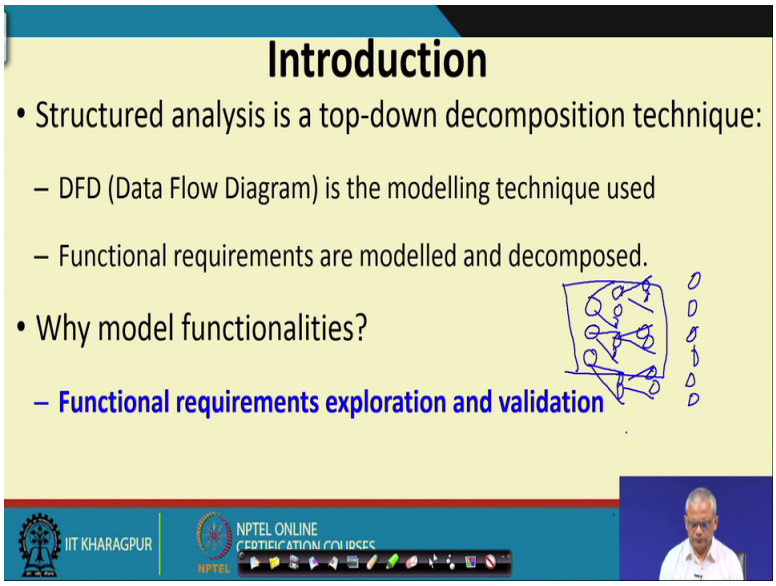
Software Engineering
Prof. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 23
Introduction to structured analysis and structured design

Welcome to this lecture. In the last few lectures we had looked at requirement specification and we saw that that is a very important issue. Before starting to develop its very important to know the exact customer requirements, find out if there are any difficult is with the requirements and then eliminate them and finally, document that in a acceptable format. And we had seen the IEEE 830 format which is accepted across all industries and academics. And we had seen the different elements of the requirements specification the functional requirements, non-functional requirements and so on and then how to document this.

Now, let us start looking at the structured analysis and design. We had also seen some very basic concepts about design namely the high-level design, detailed design and also what can be considered is a good design in terms of modularity, cohesion, coupling, layering and so on. Now, let us start to discuss how to go about doing procedural design of a problem. We assume that the requirement specification is done complete documented and we would like to start doing the procedural design for the problem. Let us see how we go about and address this issue. In this lecture and next few lectures we will address the topic of Structured Analysis and Design.

(Refer Slide Time: 02:25)



This is a top down decomposition technique, in the sense that we will start with very gross or course level requirements that is as specified in the SRS document. The functionalities that the system should perform you start from their and then we do a top down decomposition; meaning that we break the required functionalities of a system into finer levels. And for this purpose we will use the data flow diagram technique. We will use the data flow diagram technique the DFD to do the functional requirements analysis, we will model the requirements and we will decompose into smaller functions.

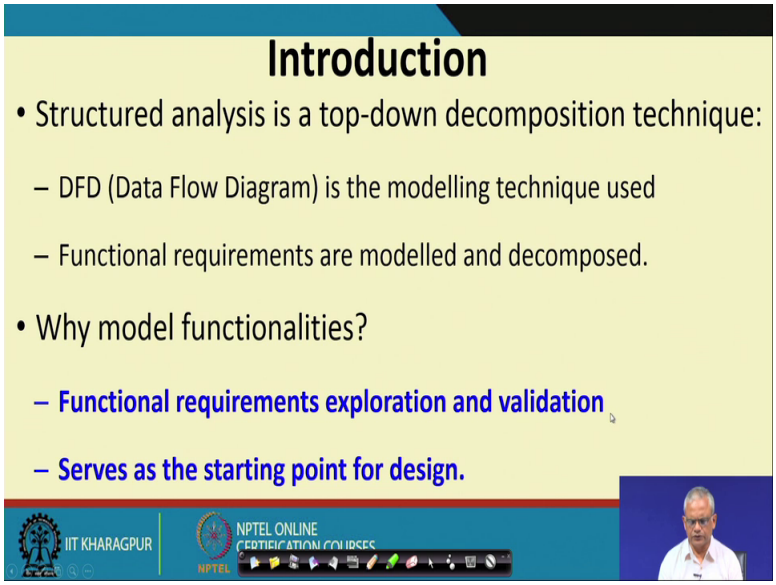
But before we started it is good to know that why we need to model the functionalities. We need to model the functionalities because we will do the requirements exploration and validation. By this we mean that the functional requirements as specified in the SRS document will take exactly the same functional requirements and we will break them down into simpler functions in a hierarchical manner.

What we mean by that is that if this is our SRS document and we have in the functional requirement section, we have various function listed there. We will take each function and we will use DFD to model this system, we will represent all these functions and also we will decompose this into finer functions, each of the functions will be decomposed.

And even the sub functions will be decomposed through the DFD technique and finally, we will have a set of fine granularity functions. So, this we call as the functional requirements exploration because base starting from very gross level requirements

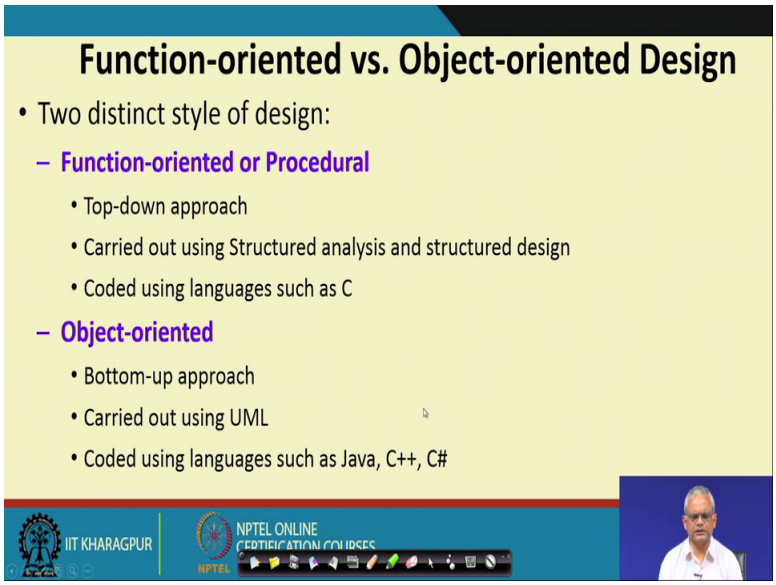
specified in the SRS document, we decompose them into functions which are very small in size.

(Refer Slide Time: 05:26)



And once we achieve this decomposition we will be ready for starting the design. So, in the structured analysis we do the functional requirements exploration that is break the functions into fine functions. And then once we have the fine set of functions, fine granularity we then start the design.

(Refer Slide Time: 05:55)

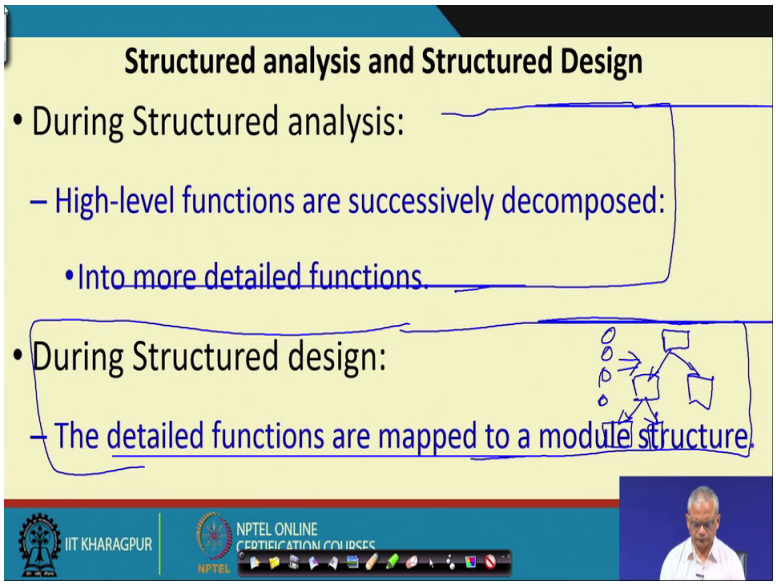


But before we start looking at the structured analysis and design we will just do a small digression, just have a small digression. We will contrast the function oriented and the object-oriented styles because right now, in this hour and next few hours we will be looking at the function-oriented design and later we will be concentrating on the object-oriented design. These are two distinct design styles. The function oriented or the procedural design is a top-down design technique as you are saying that we start with the requirements, take the functions functionality specified in the requirements and decompose them hierarchically into a fine set of functions.

And then we map them into a design structure and we use the structured analysis and structure design for the procedural design technique. And then once we have got the design then we can easily code it into using a language such as C. On the other hand object-oriented design technique is a bottom-up technique. Here based on the requirements we first identify the objects and then we see the object relations if there are any, I mean there will be many relations and then we model the object relations to build a hierarchy.

So, we start with the basic objects and then we group the objects into classes and then classes into a heritance hierarchy and so on. So, this is a big difference that function oriented is very intuitive stop down approach and object oriented is a bottom up approach a start with objects and build from there. And we will use the UML notation to do the object-oriented design that is as soon as we complete the procedural design we will get started with the function with the object-oriented design. And once we have the design ready then this can be easily coded in a language like Java, C plus or C star ok.

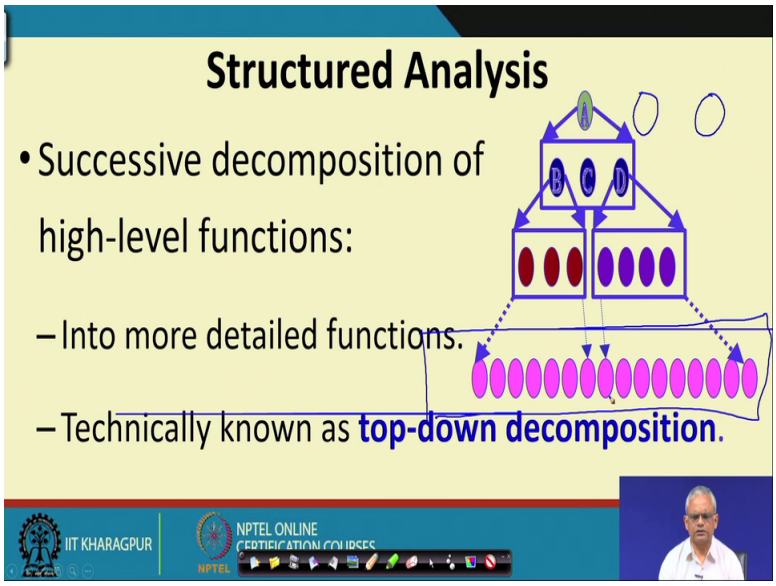
(Refer Slide Time: 08:35)



So, now let us get into our discussion on structured analysis and design. So, here as we are saying that we will start with the high-level functions as specified in the SRS document and then we will decompose into detailed functions. So, that we will call as the structured analysis. So, this is the structured analysis activity where we do a functional decomposition.

On the other hand immediately after the functional decomposition we will start the structure design and here we take the functions identified in the structured analysis. If these are the set of functions that are identified in the structured analysis; we will discuss a technique called as a structured design technique, which gives a methodology how to map these set of functions fine level functions into a module structure. And this will form our high-level design. We will first discuss the structured analysis where we identify the fine grained functions and next we look at the structure design, where we discuss how to decompose sorry how to map the fine grained functions into a module hierarchy.

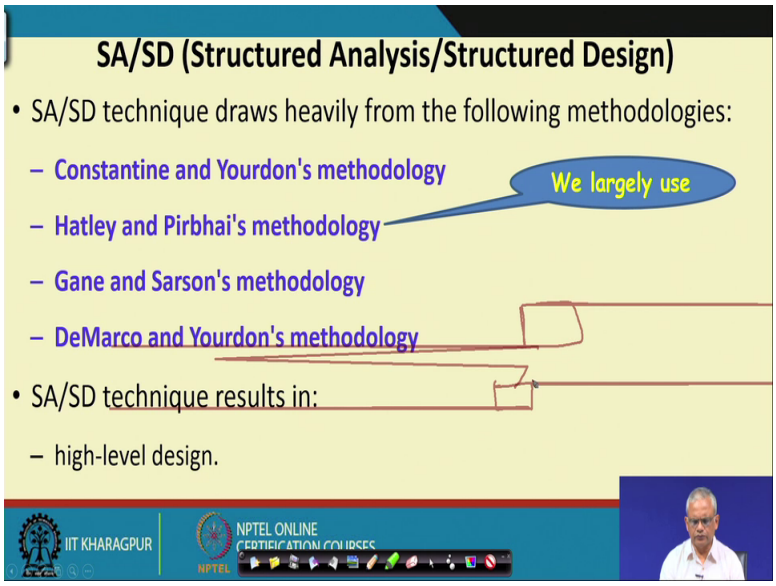
(Refer Slide Time: 10:28)



Now, for the next hour or so we look at the structured analysis where we do a top down decomposition starting with a functions identified in the SRS document, we split into sub functions and fine grained functionalities. This diagram just explains this concept, we will identify the set of functions that are there in the SRS document. If these are the set of functions for each function we will decompose that into sub functions.

How do we decompose that we will be discussing, we will use the DFD notation for modeling this decomposition. And then each sub function into further sub functions and same thing will do for the other high-level functions mentioned in the SRS document. And at the end we will have the set of fine grained functions and this we will mapped into the design structure, the high-level design in the structure as a structure design technique.

(Refer Slide Time: 12:00)



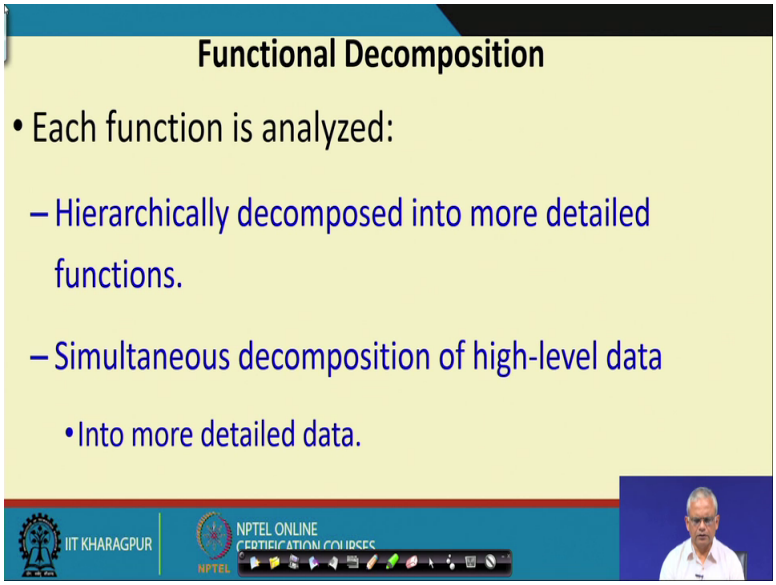
The structured analysis and structure design technique is well accepted by the developers. Anybody wanting to do a procedural design would do the structured analysis and structured design, but then there are various flavors of these design. For example, may use the Constantine and Yourdon's methodology or Hatley and Pirbhai's methodology, Gane and Sarson's methodology, DeMarco and Yourdon's methodology.

So, what I want to say is that there is no standardization of the methodologies we can find in different industries, using different methodologies. These are largely the same, but then there are small differences in the notations and so on, we will point that out. And for our discussion we will be using largely the Hatley and Pirbhai's methodology. We will see that the different tools that are available, they also support number of methodologies.

And different industry and different projects they might adopt some methodology. Hatley and Pirbhai's is a methodology which has come much after Constantine and Yourdon, Gane Sarson's. So, there is a more recent methodology and obviously, they had the privilege of making use of the ideas and the Constantine and Yourdon, Gane Sarson's, DeMarco Yourdon methodology and we will be using the Hatley Pirbhai's methodology to a large extent. And once we do the structured analysis and structure design we will have the high-level design. In the high-level design we will have the module structure

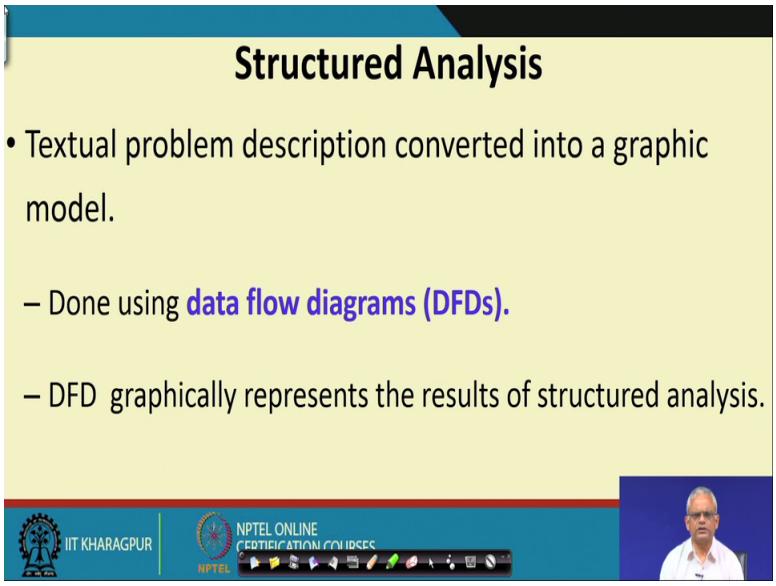
and in the module structure we can use this to code the design, to for coding and the final implementation.

(Refer Slide Time: 14:17)



Now, let us see how do you achieve functional decomposition because this is one of the main objective of the structured analysis, to do a do a functional exploration or functional decomposition. Here through a hierarchy of levels we decompose each function into sub functions and so on. And as we decompose the sub functions at the same time we will observe that the data also gets decomposed. Starting from the very high-level data that are mentioned in the SRS document, we decompose that simultaneously and automatically into detailed data. And finally, we have primitive data elements at the lowest level in terms of integer character and so on.

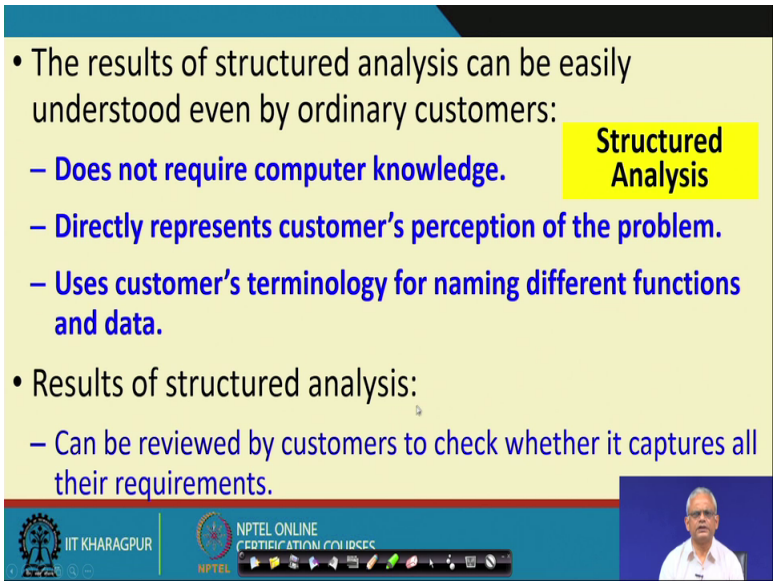
(Refer Slide Time: 15:20)



For doing the structured analysis we will use the DFD technique. DFD stands for data flow diagrams and this is a very intuitive and simple technique. It takes hardly few minutes to get familiar and get productive and get started using this technique.

And because of its simplicity and utility this technique is extremely popular not only in the procedural design, but also in many other applications. For example, in the SRS document itself in many projects they develop the DFDs because this gives a good understanding of the functions into a finer level of activities. And also the DFD model is a very intuitive model and by just looking at it we get a good idea about what the system needs to do.

(Refer Slide Time: 16:35)

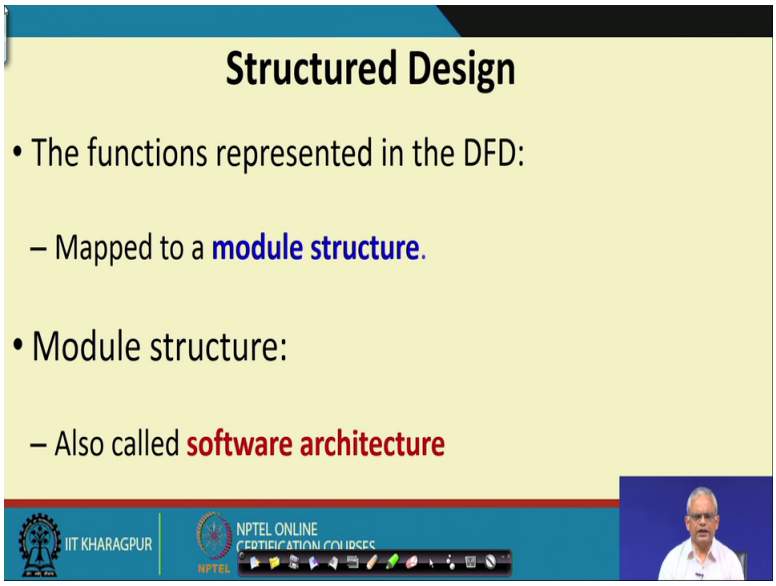


As we were saying that the structured analysis technique is a very simple technique, do not need to be even knowledgeable in computer; anybody can start using this technique. And it represents the customers perception of the problem, this is not really a design technique it just keeps on elaborating what the customer wants at the problem. So, this is a detailed model of what the customer really wants of the problem and for this we also use the customers terminology for giving various names to the data to the functions and so on.

And no wonder that this technique is easily understood. The model is developed using the DFD are easily understood by the customer even though customer may not have computer exposure. Therefore, once the designers they come complete their structural analysis the DFD model of the system, it can be reviewed by the customers to check whether it missed any of their requirements and whether it has correctly captured their requirements. Based on this idea that this is a very simple technique can be learnt in just few minutes. We can learn this technique and start modeling various problems in a very short time.

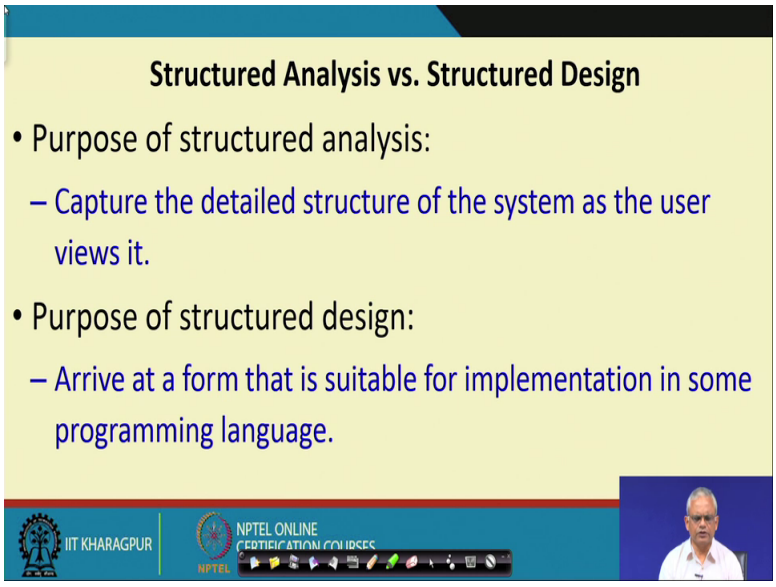
But let me just tell you that it is easy to learn this technique, but then for finally, using this technique to model various problems we need bit a practice. We will define a set of assignments, you can carry out the structured analysis model of this technique using DFD and then check whether you are doing it right.

(Refer Slide Time: 19:03)



And once we have completed the structured analysis we will start the structured design. And here we will map the fine grained functions into a module structure and this module structure is also called as the software architecture.

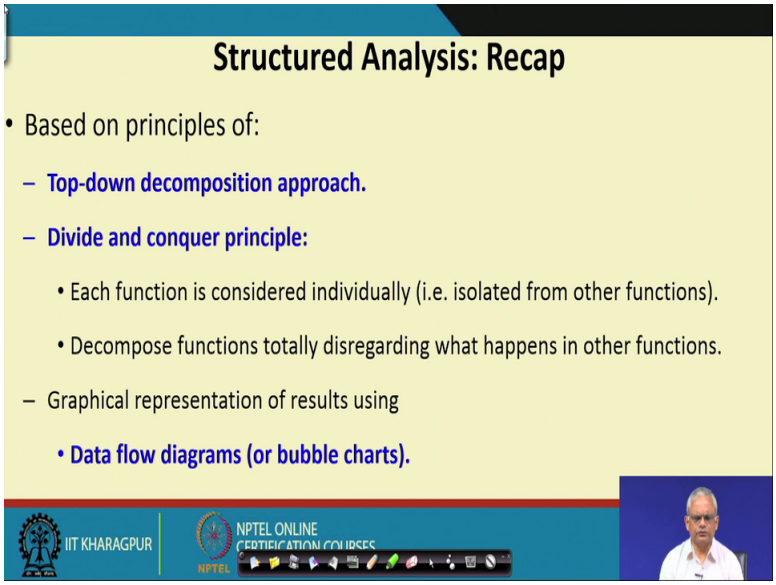
(Refer Slide Time: 19:21)



One thing we must be clear is that the structured analysis is just a elaboration of the things that the customer wants. On the other hand in structured design we arrive at a form that can be easily implemented that is coded into a solution. Obviously, the form structured analysis if you are done the structured analysis we cannot derive a code from

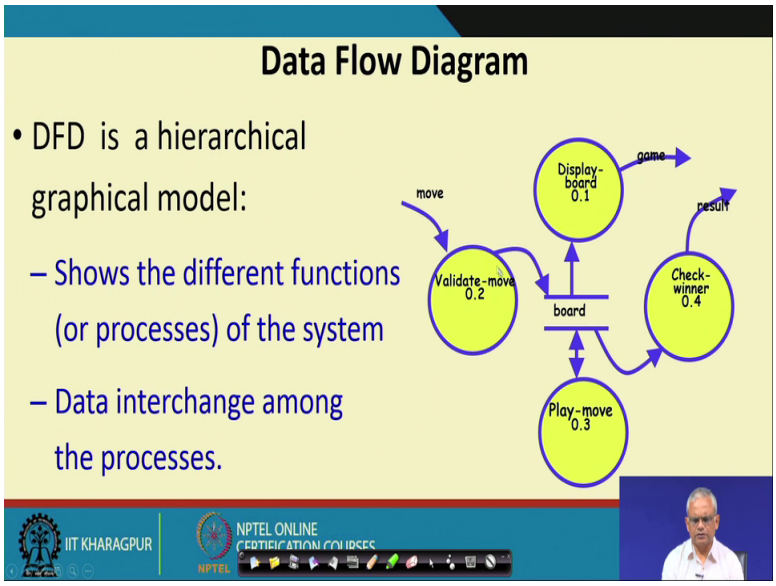
that, but once we have converted the structured analysis into a high-level design using the structured design technique that can be easily coded into the solution ah.

(Refer Slide Time: 20:17)



Just to recapitulate what we discussed, the structured analysis is a top-down decomposition approach. We know what we mean by top down decomposition and it is a divide and conquer principle. It uses the divide and conquer principle because, we look at all the functions in the SRS document, but take each function one by one and then at a time we decompose one function. And then we decompose it into sub functions and again at a time we look at one sub function and again decompose it and that is exactly is the divide and conquer principle.

(Refer Slide Time: 21:14)

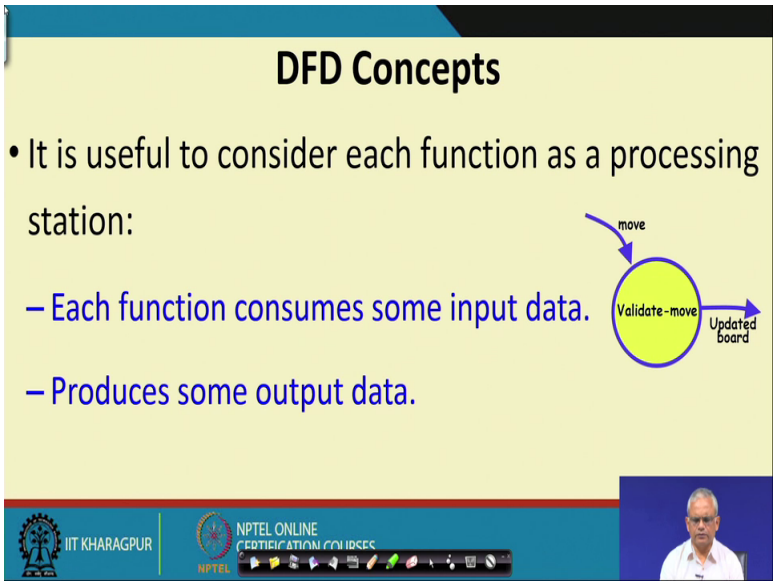


And it is a graphical technique, we will use the DFD which is also called as bubble chart to represent our result. The DFD allows us to do a hierarchical decomposition and that is why the DFD is a hierarchical graphical model, we will have various levels of model. Some are very high-level model and then these are further decomposed into finer or more detailed levels.

And at each level we will represent the functions or the processing in the system and also the data interchange among the processes. This is an example of a DFD model, we will use the circles to represent the functions of the processes that is the terminology. Either we can call these as bubbles, processes or functions and then this takes some input data produced some output data because this is the data flow a model. So, it represents how the data flows within the system through the various functions.

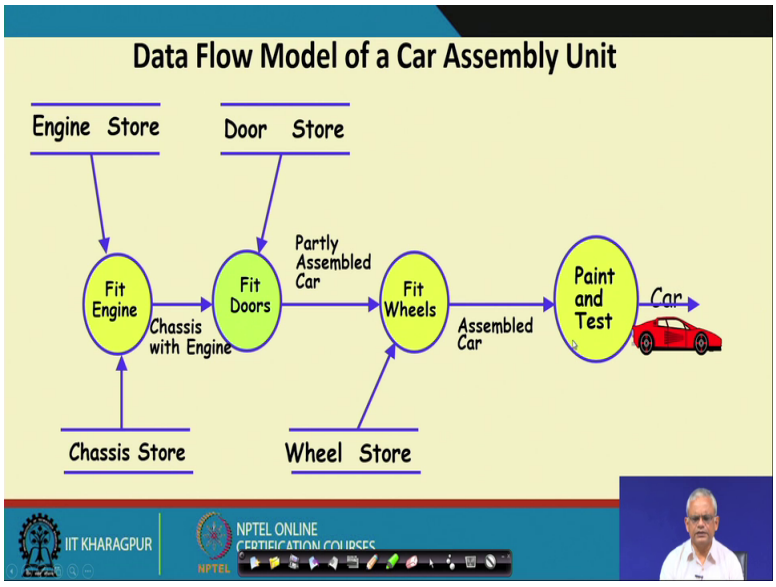
Please remember that this is not a control flow model, it does not say that which flow occurs first and then which one occurs next and so on. It just represents the data flow basically, what are the functionalities at this level and what data they consume and what data the produce. We will see the other notations that are used here in few minutes.

(Refer Slide Time: 23:10)



As we were just saying that each a function is represented using a circle and this represents some processing. It takes some input data, does some processing and produces some output data.

(Refer Slide Time: 23:37)

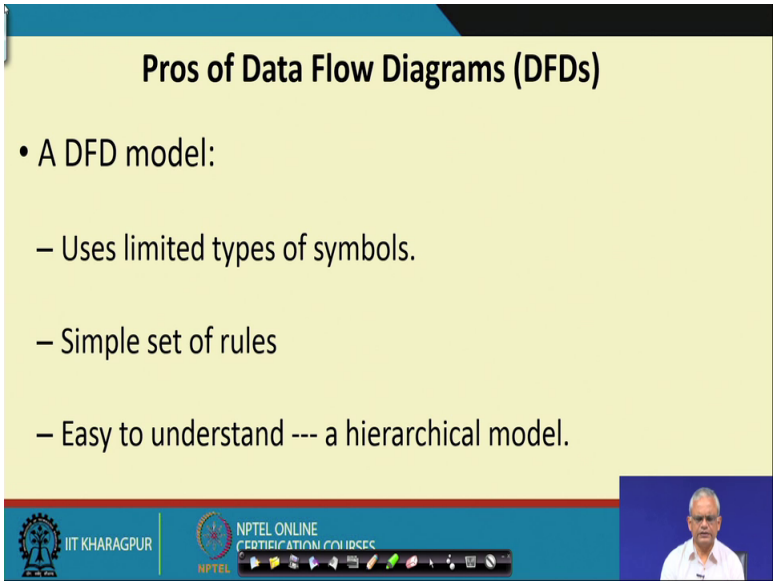


The model is extremely simple as you are saying without much effort we can easily understand a model developed using this, just let us look at this car assembly unit modeled using the DFD notation. So, here the processes fit engine, fit door, fit wheel paint and test; you can observe here that each of the process represents some activity and

that is why these are starting with the verb form fit engine, fit door, fit wheel, paint and test.

So, every function represents an activity or processing and is named using a verb form that is doing something. And then each processor bubble here it takes some data, it takes engines from the engine store, takes chassis from the chassis store and then fits the engine to the chassis. The chassis with the engine goes to the next processing station and here the doors are obtained from the door store and to the chassis with engine the doors are fitted. And this we call as the partly assembled car and then to this the wheels are fitted by taking the wheels from the wheel store and then we have the assembled car. This is painted and tested and the final output is the car.

(Refer Slide Time: 25:43)



So the model is very intuitive, just by looking at the model we can make out that what really happens. It is a very simple technique, as you are saying we will see that in very small amount of time. We can start doing the DFDs for various problems. One of the main reason why it is very simple technique is there are only 5, 4-5 types of symbols actually 5 symbols to be precise; we can learn the 5 symbols in no time. The set of rules by which we combine this symbols to model, a system very simple and also it is a hierarchical model. We start doing something very simple and slowly add more details to it and therefore, while even developing a very sophisticated model we find that it becomes easy.

So, with this discussion about the DFDs structured and introduction to structured analysis, we will stop here. We will continue from this point in the next lecture. We will see how to use the DFD technique, what are the nitty gritty of the DFD technique and how to use this to model any given problem.

Thank you.