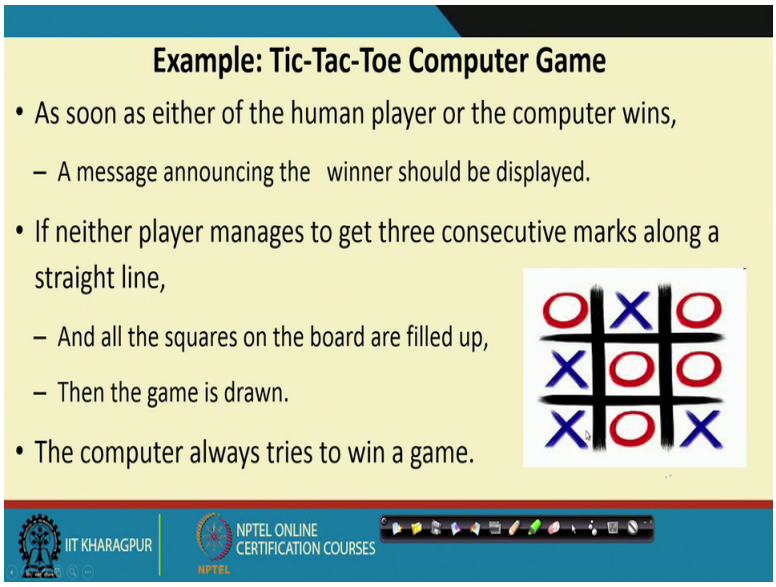
Software Engineering
Prof. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 26
Examples of DFD Model development

Welcome to this lecture. We had discussed about the DFD model, the symbols that are used, how to draw the different levels of the DFD diagram and we had also looked at some very simple problems. In the last lecture towards the end we are trying to develop the DFD model for the tic-tac-toe problem, the tic-tac-toe is a simple computer game.

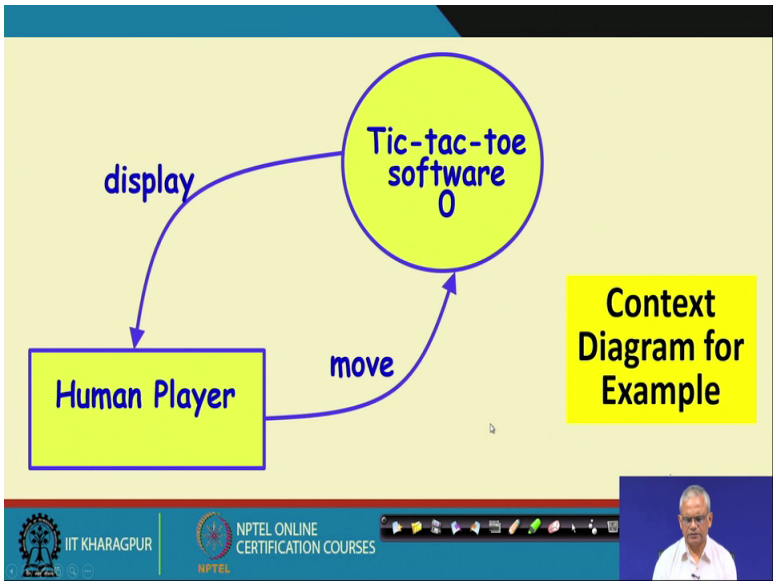
(Refer Slide Time: 00:51)



And here there is a 3 by 3 square where the computer and the human player they take turns to mark 1 square and whoever gets three consecutive marks, on either a diagonal a row or a column wins the game and if all the squares get filled up without anybody getting three consecutive symbols in the game is considered drawn.

Now, we want to develop a software for this and we want to develop the structured analysis document and the DFD model for this. The first thing to do is the context diagram and the context diagram is extremely simple. Well, we draw 1 circle, write the name of the software there and then identify who are the users. There is only one user and that is the player, the player enters move and receives the response.

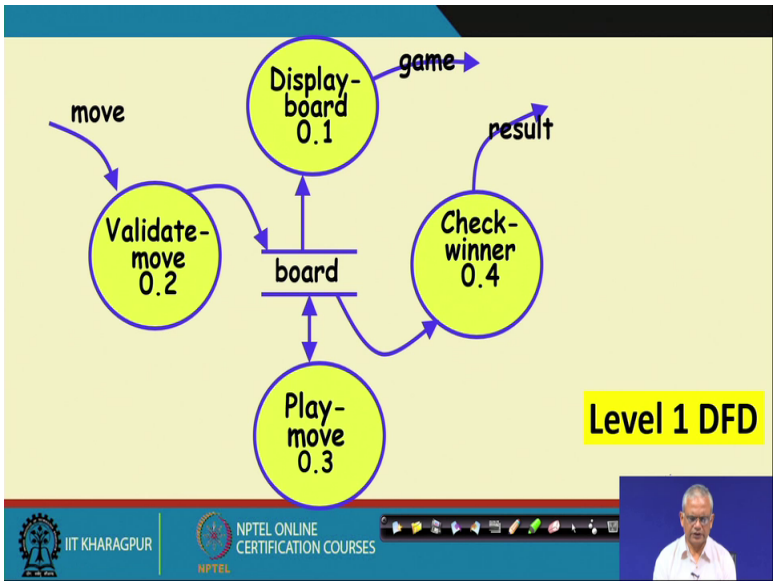
(Refer Slide Time: 02:16)



So, that is the context diagram very simple here, that write the name of the software Tic-tac-toe software, number is 0. The external entity is human player, enters, moves and receives the display. Now, how do we develop the level 1 diagram? In the level 1 diagram, we identify the high-level functionalities and represent them as bubbles typically between 3 to 5 bubbles in the level 1 diagram.

If we look at the problem description one is about reading the move and validating that. The other is about playing the computer move, the computer move need to see the board position and then need to identify what the; would be the move for the computer and then display those.

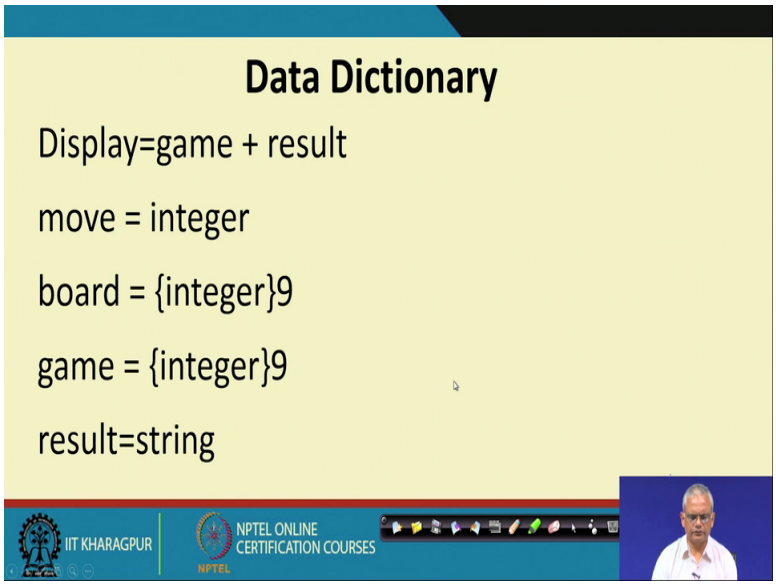
(Refer Slide Time: 03:45)



So, we can model that is validate move once, the move comes validate move and then the move is marked on the board display board and then the computer needs to play move and the check winner. So, once the human player makes the move; the move is validated that is updated on the board. And then the check winner must be there just to check whether, after the human player has played a move if he the player has won. If won then the result is displayed otherwise, the computer makes the move. Again, the winner is checked and the display board is made.

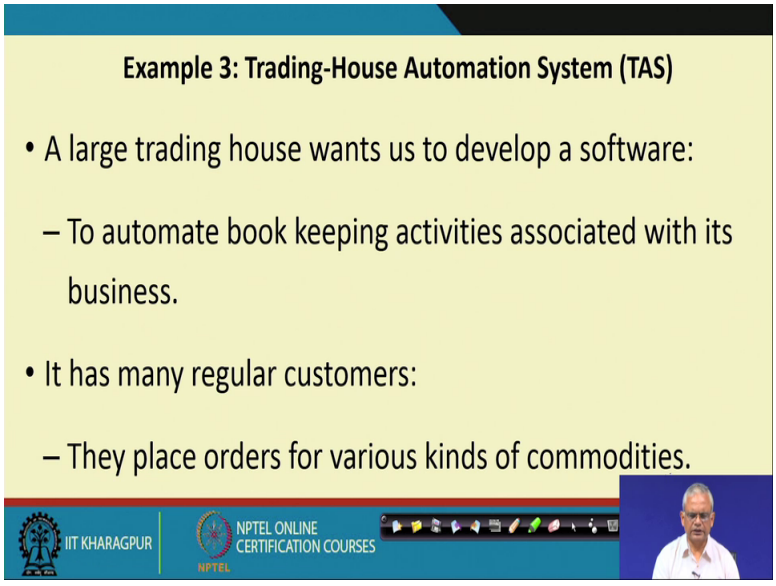
But just remember that this is the data flow model and therefore, we do not represent control aspects like which bubble will operate after which one. Those are not really important to our design that which one will execute after which one. We are just identifying what are the sub functions of the play move and sorry the handle move a move is made by the human player. What are the sub functions of the handle move and then we just represent them here.

(Refer Slide Time: 05:41)



Now, we develop the data dictionary right. The display is game plus result because that will result in the balancing of the diagram. The level 0 diagram, the context diagram if you look at then displays the data that is produced by the system whereas, in the level 1 diagram there are two data that are produced game plus result. Therefore, to balance must right displays game plus result, move is an integer, the board is 9 integers and the game is also 9 integers and the result is a string.

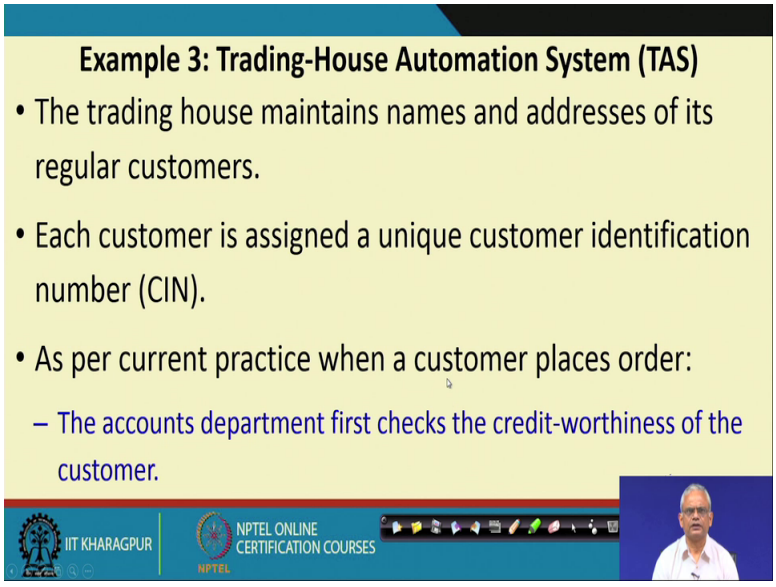
(Refer Slide Time: 06:36)



Even the tic-tac-toe problem was a very simple problem. Now, let us do a slightly more sophisticated problem which is the Trading-House automata an Automation System.

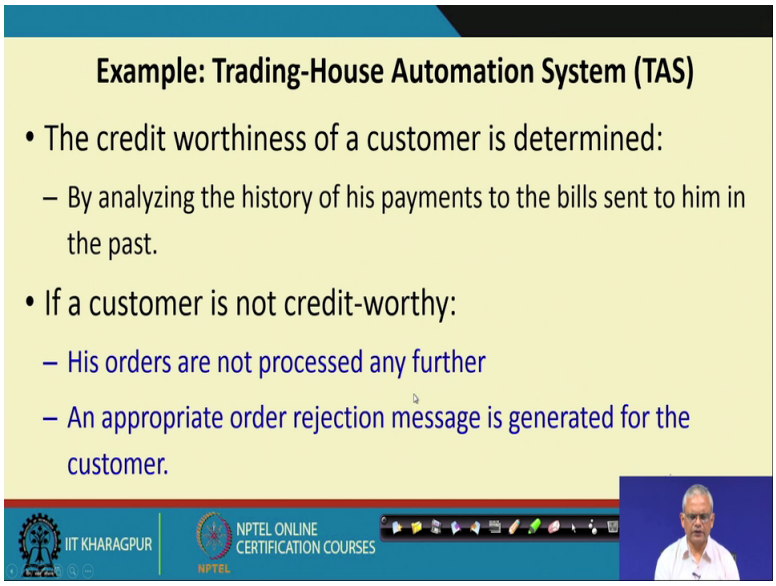
In this there is a trading-house which wants us to develop a software to automate its the bookkeeping activities. The trading house has many regular customers who place orders or various kind of commodities with it.

(Refer Slide Time: 07:11)



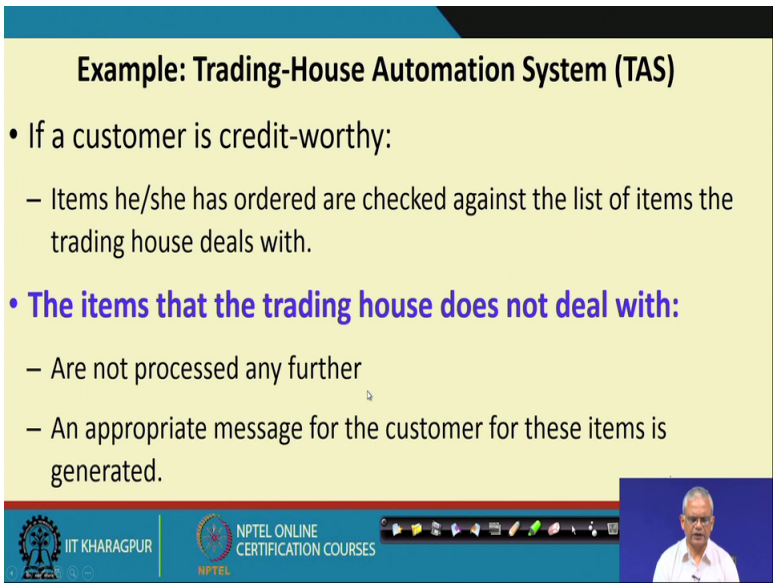
Now, the trading-house has a set of registered customers called them as regular customers and each customer has a identification number. And when one of the registered customers places order then the account department checks the creditworthiness of the customer.

(Refer Slide Time: 07:40)



The creditworthiness is checked by finding out the payment history of the bills sent in the past. If the account department says that the customer is not credit worthy then, the bills are his orders are not processed further and the rejection message is issued that sorry we will not be able to handle your request.

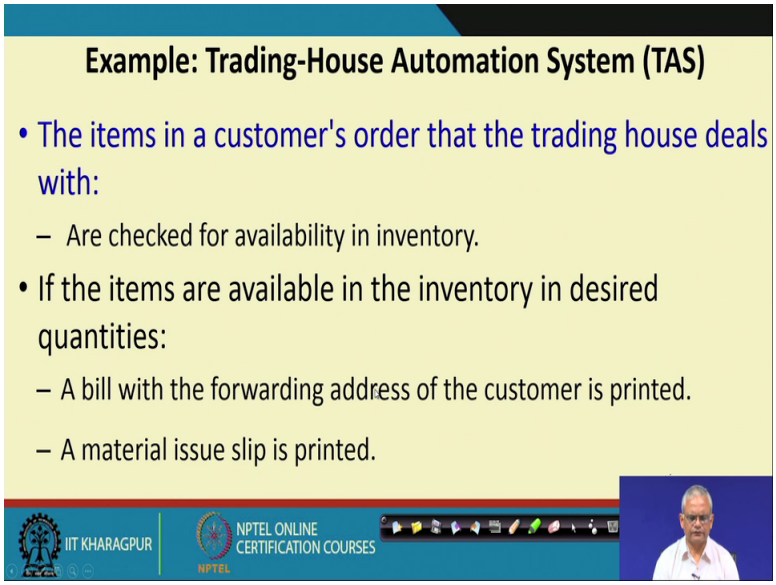
(Refer Slide Time: 08:07)



But if the account department says that the customer is credit worthy then the orders are checked against the list of items that the trading-house deals with. For those items who is

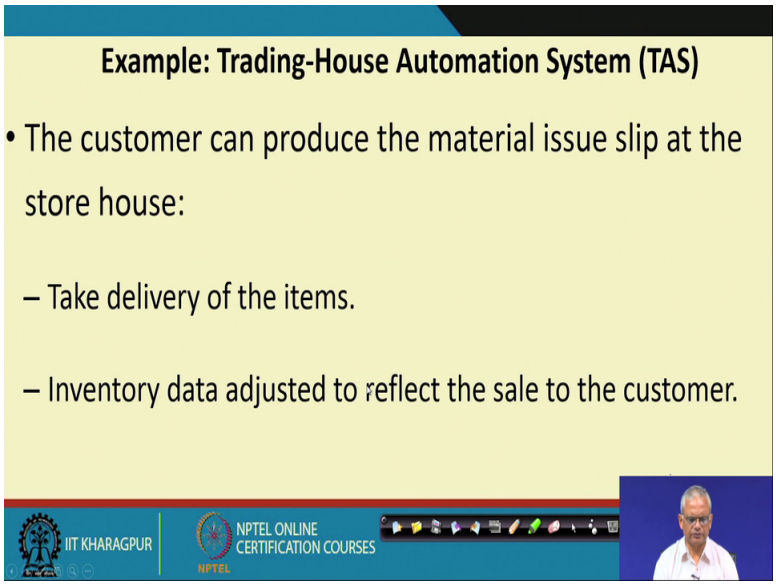
the trading-house does not deal with it just does not process them further and just sends a message saying that it does not deal with those items.

(Refer Slide Time: 08:32)



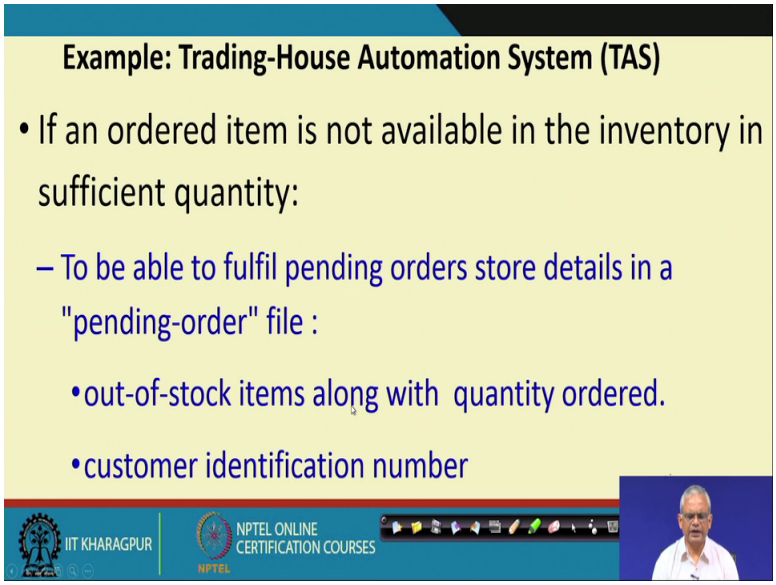
And for those items that the trading-house deals with, it checks its inventory. If the items are available in the desired quantity then a bill is printed and sent to the customer. And also a material issue slip is printed, the customer can present the material issue slip and take possession of the items.

(Refer Slide Time: 09:04)



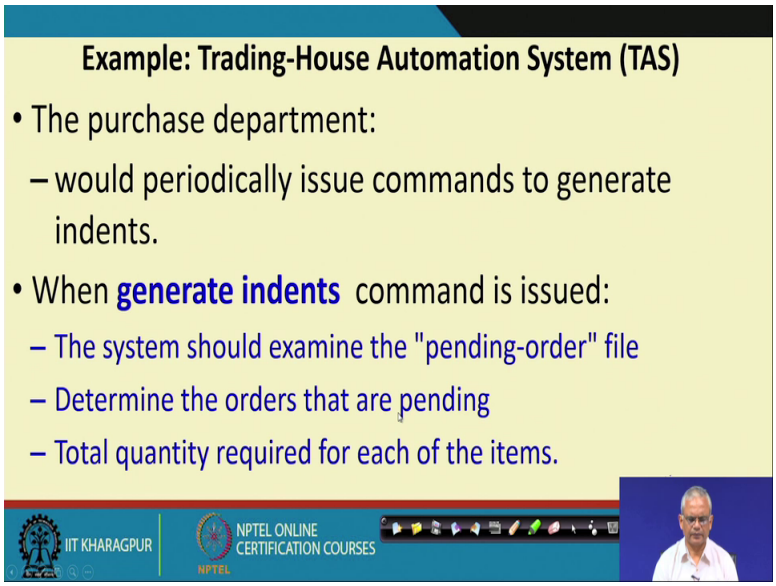
And once the items are been, bill has been raised for a certain item then the inventory data is automatically adjusted.

(Refer Slide Time: 09:21)



And if some item is not available in the inventory then there is a pending order file for that item is created. And the number of items the quantity that is required is stored there along, with a customer identification number who has raised that request.

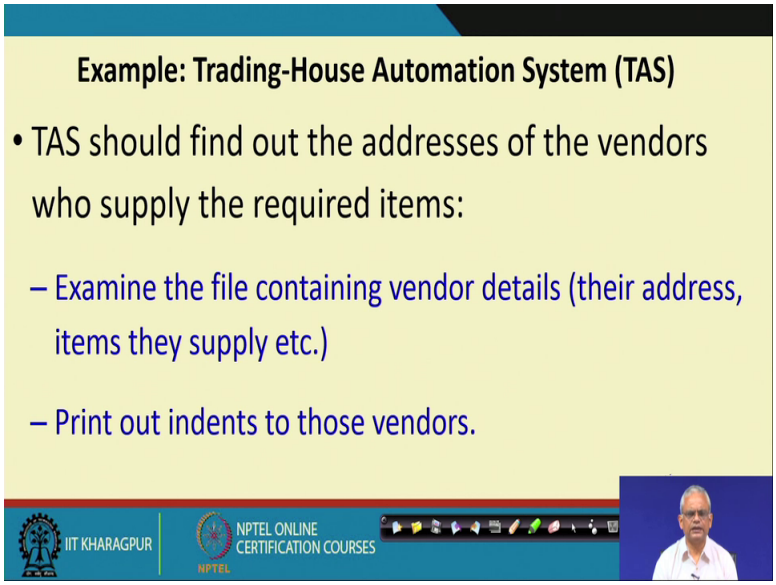
(Refer Slide Time: 09:43)



The purchase department periodically issues commands to generate the indent. And here for those items which are out of stock the number items, number of items that are already

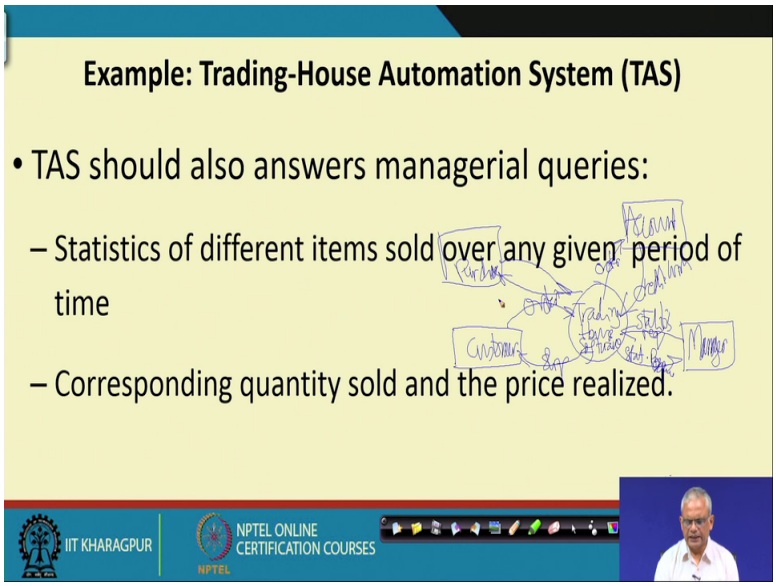
demanded by the customer for that item is taken into account. And the required quantity for each item is placed against the vendors who supply those item.

(Refer Slide Time: 10:23)



The vendor details are maintained in a file, their contact address, the items the supply and so on. And for a specific item the indents are raised against the specific vendor who supplies that item.

(Refer Slide Time: 10:45)



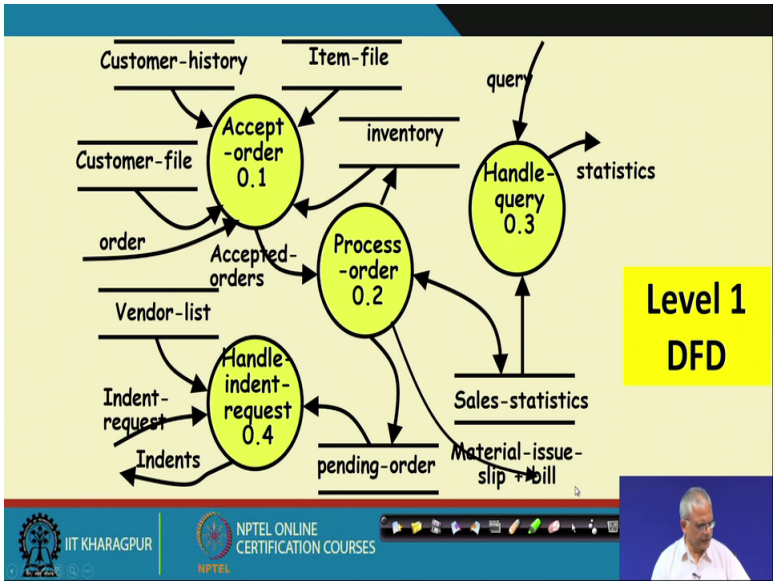
Other than handling requests from the customer it also answers managerial queries like; What is the statistics sales statistics? What is the volume of items sold? What is the price

realized? And so on. So, if we want to do a structured analysis of this we need to first do the context diagram. And context diagram consists of drawing 1 circle with the name of the software in that and identifying the context in which the system exists. That is the different users who would use the software, what data they would input and what data they would receive..

And here if we read the problem, we will find that for this system the name of the software is trading-house software. Just write that here trading-house software and then we identify that there is the customer is one of the user and the manager is another user. We have the account department as an user and we have the purchase department who raised indents they are also another user.

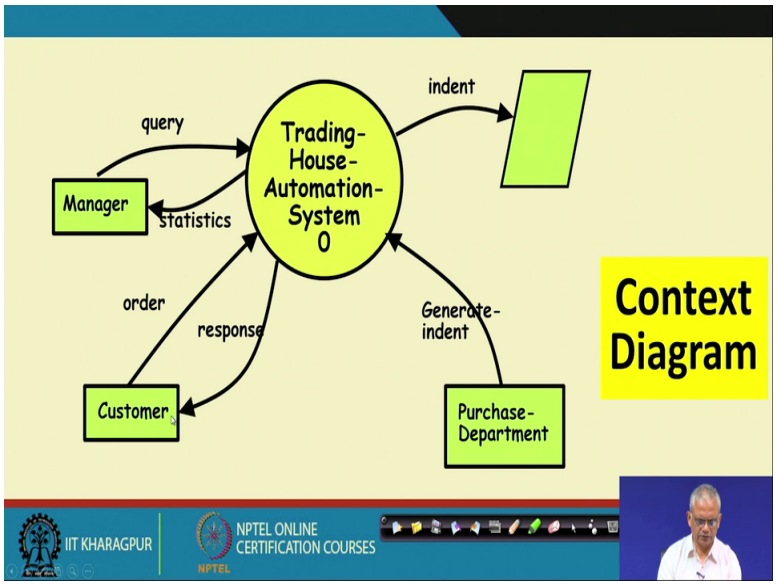
The customer places orders and receives the material issue slip. The account department checks the creditworthiness, looks at the order and checks the creditworthiness. The purchase department looks at the pending orders and issues the indents. The manager requests for statistics, statistics requests and gets the statistics response, statistics request and gets the statistics response.

(Refer Slide Time: 13:47)



So, drawing the level 1 diagram is rather straightforward. So, we have drawn the level 1 diagram here.

(Refer Slide Time: 13:49)



The customer, the manager not shown the account department here, but the account department also ok; so, sorry the account department is software that is inside it, the accounts department role is done by the software. So, not shown the account department as a different entity here. Automatically that is done here by looking at the credit worthiness here and therefore, no specific extra input required by the account department..

It is done within the software, the creditworthiness of the customers is checked. So, the customer places orders and receives response which is the material issue slip and also item not dealt with by the trading-house system etcetera, those messages. The manager can raise statistical queries and get statistics. The indents are placed on the vendors and the purchase department places, generate indent..

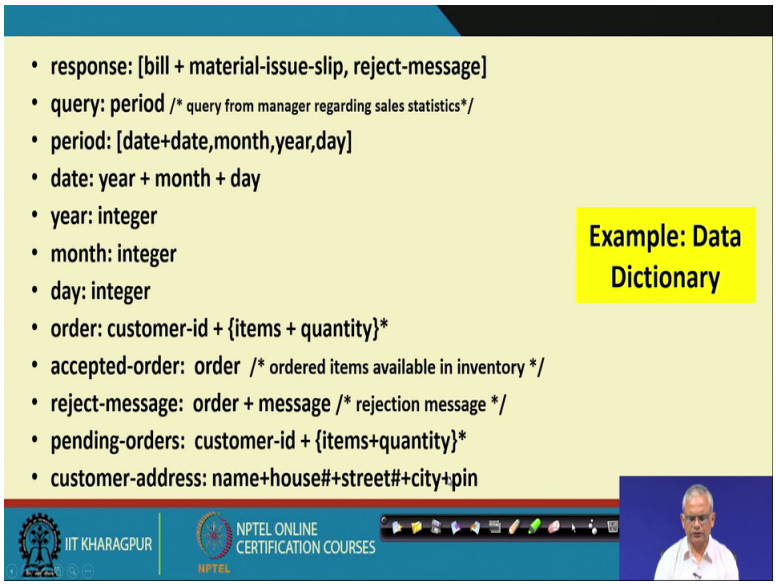
Now, the level 1 DFD if we read through the problem description and identify the functional requirements; we will see that the major functional requirements are the high-level functions, there are accept order. So, the order is placed by the customer and then it checks whether, it is a valid customer by checking at the customer file. And also checks the customers payment history, automatically decides whether the customer is credit worthy.

And then checks the item file, if the item is actually dealt with by the trading-house. It also checks the inventory file and the accepted order is given to the process order. The

process order looks at the inventory, if they are available then it would issue a material issue slip bill and it would update the sale statistics and also update the inventory.

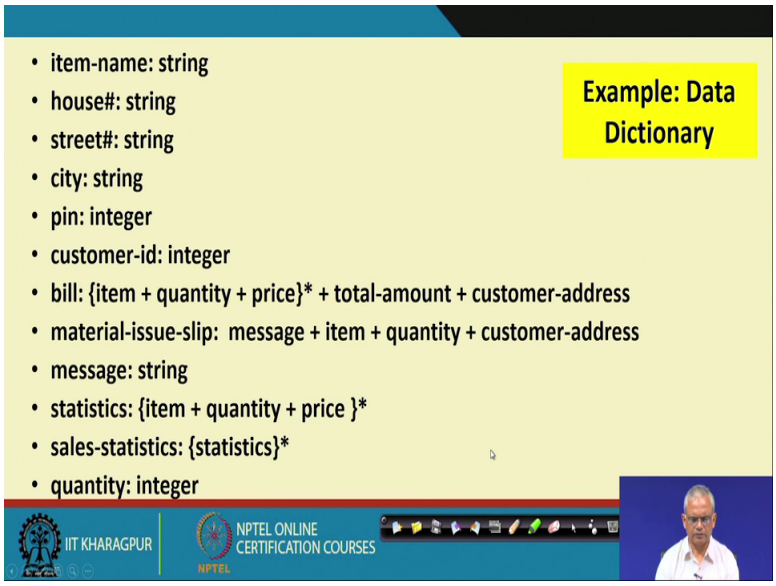
But if the item is not available then it will record that in the pending order file the inventory does not have enough item, it will update the pending order file. And sometime the handle indent requests will be operated by the purchase department. They will give indent request and then against the vendor list and the pending order the indent will get generated. The managerial queries concern only the sales statistic and that is once the query comes the appropriate statistics are obtained from here and displayed. So, that is about the level 1 DFD.

(Refer Slide Time: 18:01)



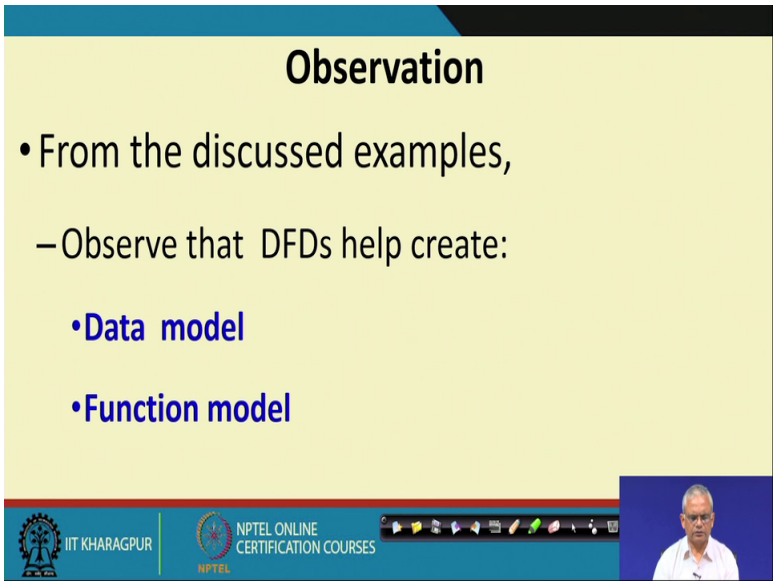
We then need to do the data dictionary. Every data item that is appear in either in the level 0, level 1 we just write the data item and for the primitive items we just directly write what purpose they are used and for composite items we write the component items.

(Refer Slide Time: 18:31)



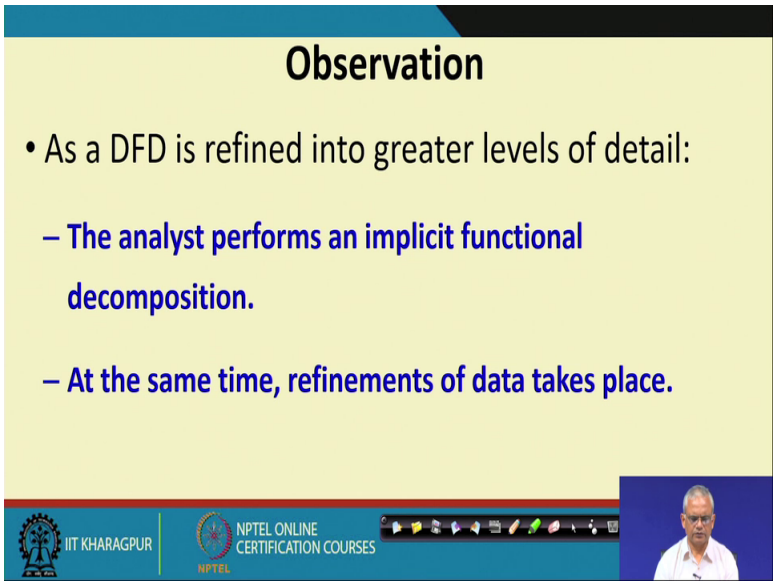
So, this again is the data dictionary which is becoming quite long.

(Refer Slide Time: 18:41)



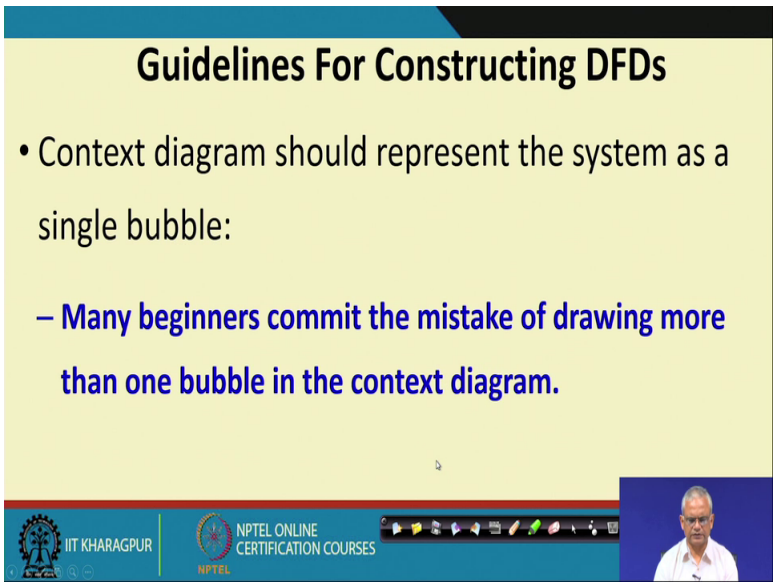
So, based on the problems that you have so far modeled using the DFD; we see that DFD is a very simple model. And we helps us to achieve decomposition, we perform a detailed function model using DFD and at the same time a detailed data model is also built.

(Refer Slide Time: 19:11)



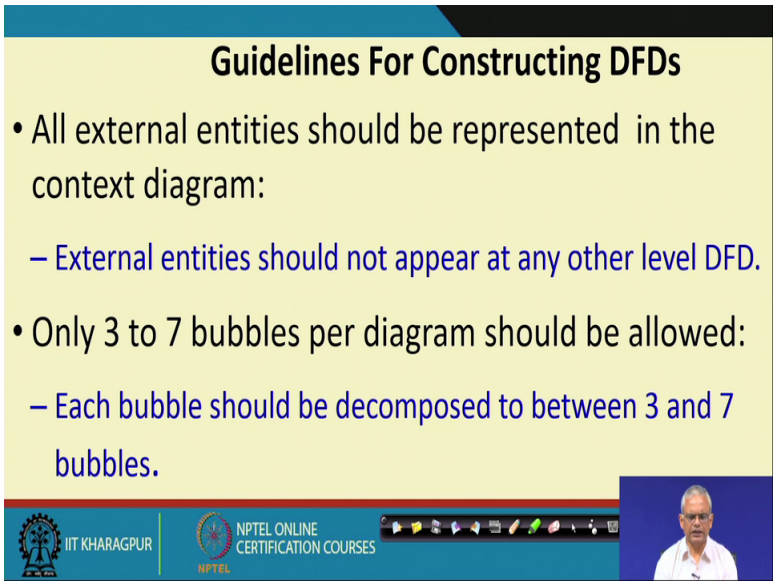
The DFDs help us to achieve functional decomposition and at the same time decomposition of data takes place.

(Refer Slide Time: 19:24)



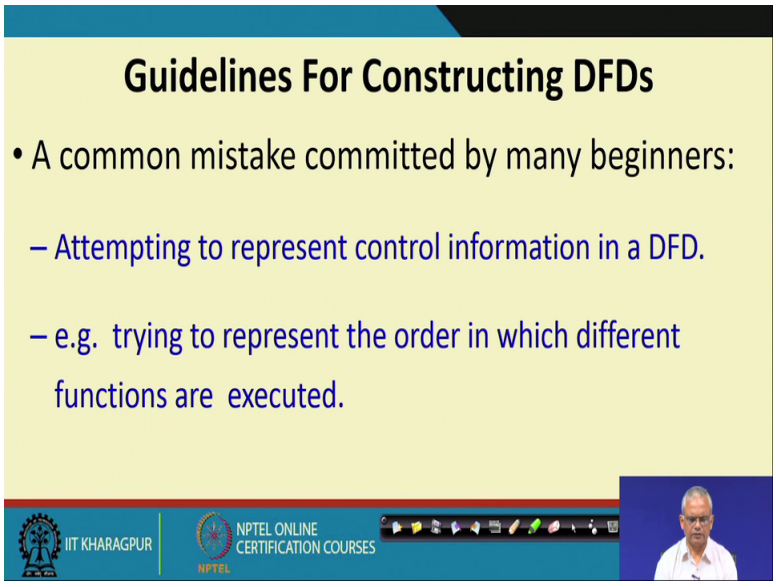
Now, let us see some of the mistakes to avoid while constructing a DFD. The context diagram contains only one bubble, that is the simplest representation. If we draw more than one bubble in the context diagram that is not correct.

(Refer Slide Time: 19:46)



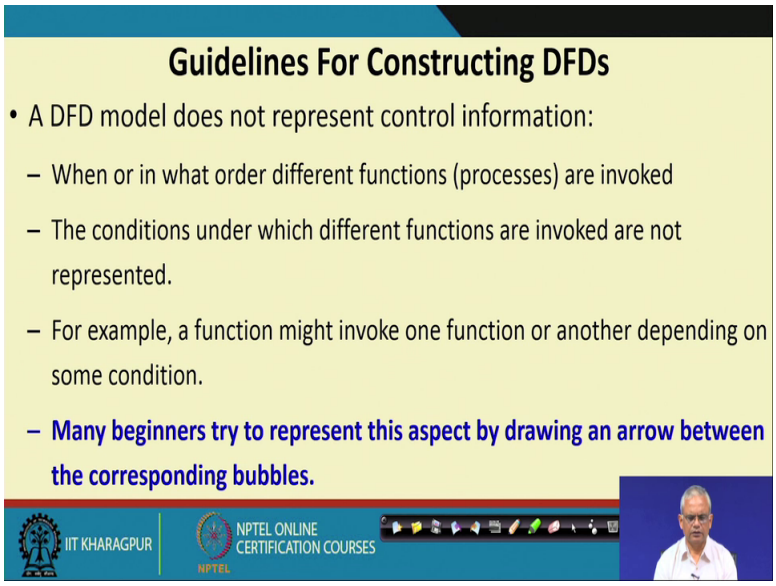
The external entities appear only in the context diagram and in level 1, level 2 etcetera context, the external entity should not appear. In any DFD level only 3 to 7 bubbles per diagram should be allowed.

(Refer Slide Time: 20:10)



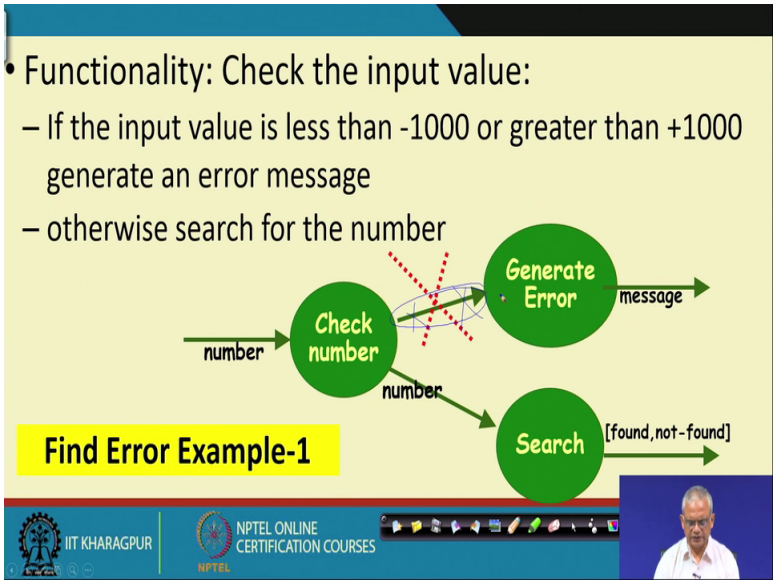
Here only the data flow is represented, we should not represent things like which one which function operates before which other function etcetera. We just represent here each function, what data it needs and what data it produces.

(Refer Slide Time: 20:34)



This is one of the very common problem. They represent the control arrows here, that is which one operates after which one and so on and that is not correct..

(Refer Slide Time: 20:58)

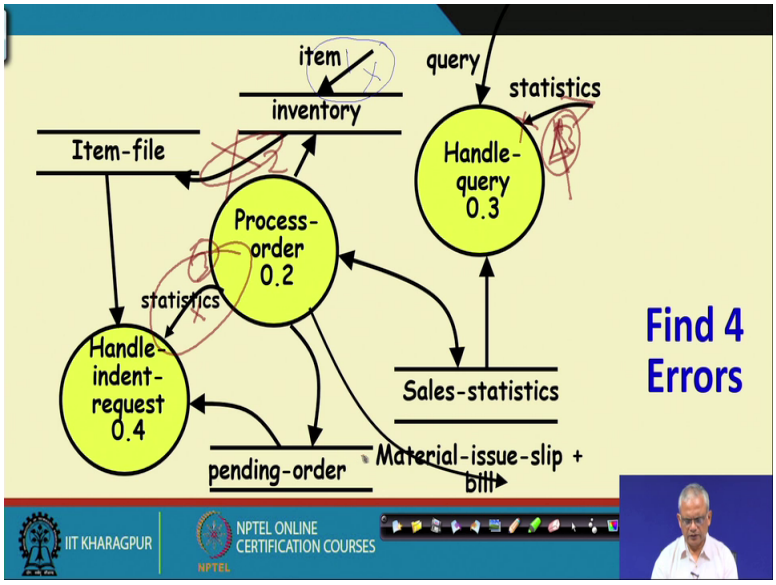


Just to give an example, let us say we have a developer who has drawn this diagram that check number. And for check number, if the number is valid that is between minus 1000 and plus 1000 then the number is searched here. So, search is another function the number is passed on to the search and which generates one of these messages found, not

found. Whereas, if it is not a valid number it generates an error message and that is the function to generate the error message produces the message.

But then I have drawn an arrow here, but generate error does not need any message any data and that is the reason why we are not able to write any data here. Then it is actually representing control information because it does not represent data, it just represents that generate error will work after check num number is completed and that something wrong. So, this arrow should not be there. It represents only control information, no data flow is indicated by this arrow and that is a mistake.

(Refer Slide Time: 22:41)

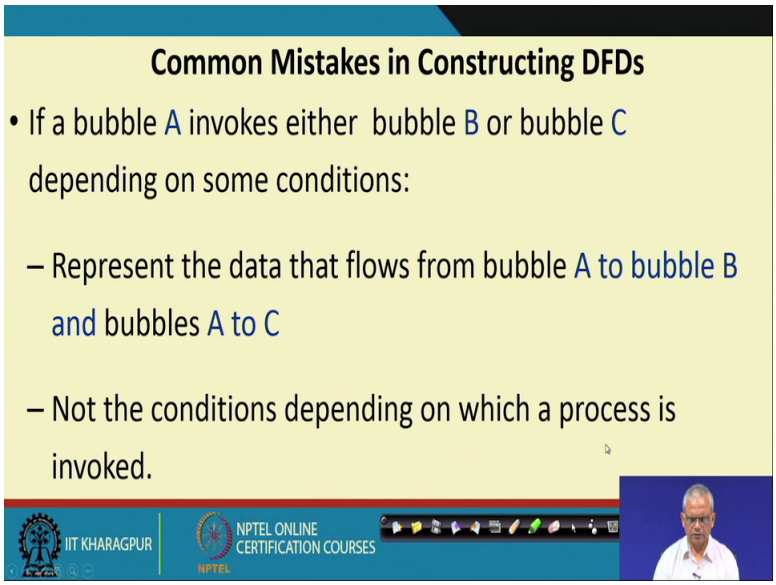


Now, let us say this is the diagram developed by somebody. Now, please find errors here. There are 4 errors here. Please observe the diagram, the first error that you can find here is that data store can be updated or read by a process. So, it cannot directly come into a data store. The second error so, this is the first error here..

Second error here is that data cannot flow from one store to another automatically, there has to be a process through which it flows. So, this is also a mistake second mistake. The first mistake is need a process to update the data store, data cannot flow from one store to another store automatically without a process. Now, are there any further errors? There are many other errors for example, the process error enters the statistics, but then statistics is not really required by handle indent request..

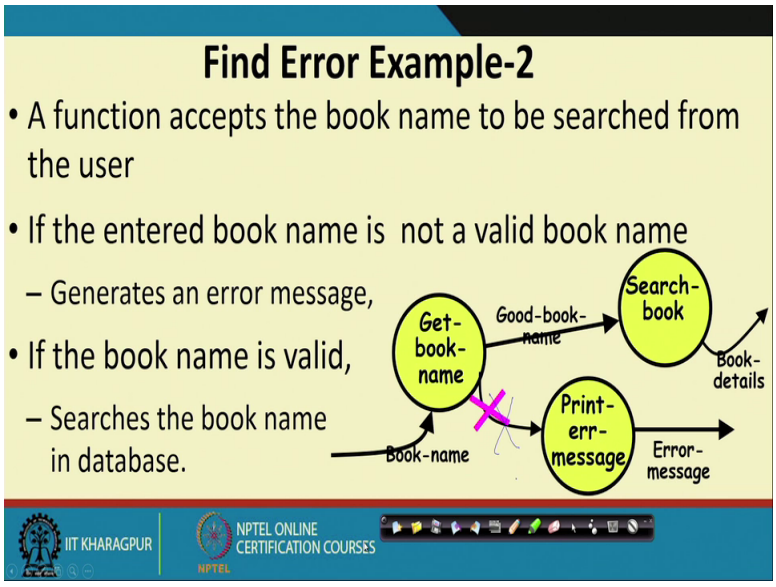
So, this is extra data that is not needed there and similarly handle queries should produce the statistics, not consume the statistics. So, the direction of this arrow is wrong. So, this is the 3, this is 4 4th mistake ok. So, let us proceed further. The common mistakes more mistakes are committed by beginners.

(Refer Slide Time: 25:13)



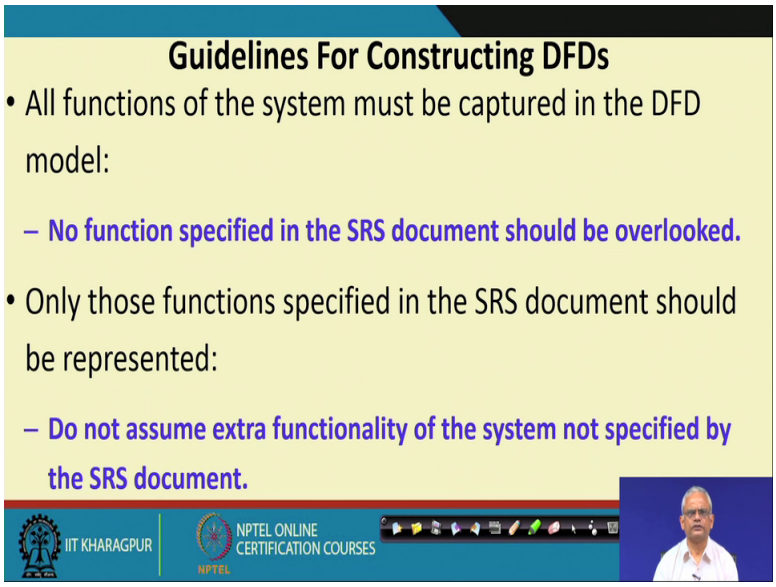
For example, if conditionally something is done then represent a arrow there and that is not really required because, we should represent only data flow..

(Refer Slide Time: 25:40)



Just to give an example, let us say this is the get book name process, reads the book name and if the name is proper; it passes the book name to the search book which produces book details. But if the book name is not proper it should print error message. So, this arrow is incorrect because print error message produces some standard error message, it does not need any input data. And therefore, this is a control arrow and this is not really required.

(Refer Slide Time: 26:47)

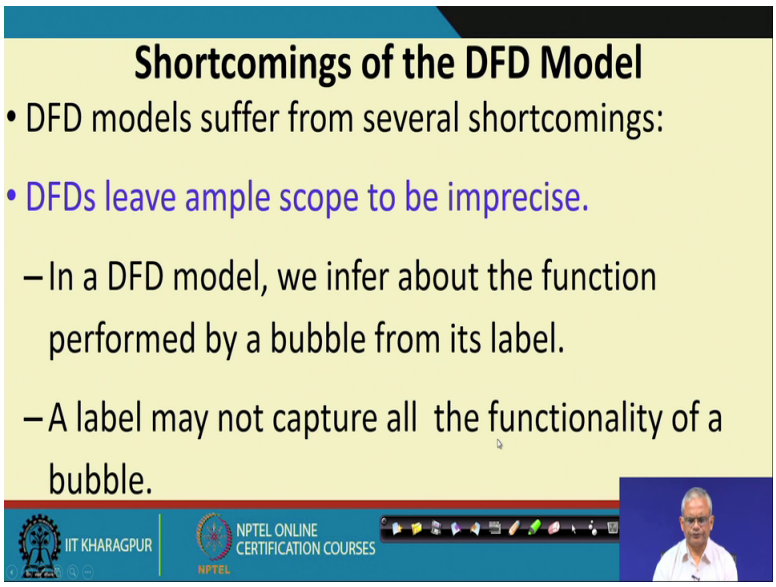


Now, let us proceed further. We should not miss any functions in the SRS document. If there are some functionalities that are mentioned in the SRS document, they must have been represented in the DFD. And it should also not represent extra functionality, which is not required in the SRS document..

(Refer Slide Time: 27:10)

Listed the commonly made errors here unbalanced DFDs, forgetting to an write the name of the data on a data flow. Functions that should be there based on the SRS document are not present or data that are should be produced or consumed are not there omitted, that is a error. External entities appearing at higher level DFDs, representing control aspects, context diagram having more than one bubble. Bubble decomposed into too many bubbles at the next level. Terminating the decomposition too early or too late, using nouns to name the bubbles, etcetera. These are some of the common errors.

(Refer Slide Time: 28:12)



Now, we will conclude the DFD model in the next lecture, just have only few items left on the DFD model. And then we will look at how to convert the outcome of the structural analysis that is the DFDs into the structured design which will give us the high-level design and which can be coded easily; that we will discuss in the next lecture. We will stop now.

Thank you.