Principles of Engineering System Design
Dr. T. Asokan
Department of Engineering Design
Indian Institute of Technology, Madras

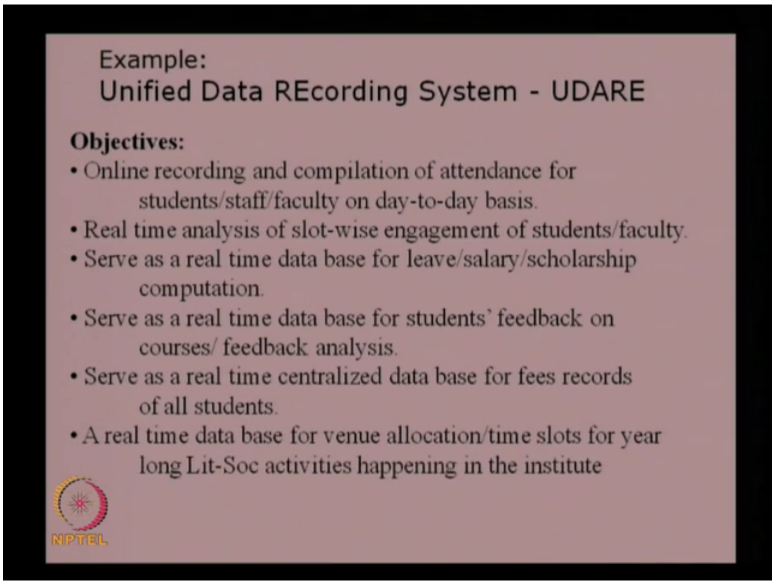
Lecture - 11
Functional Decomposition: Examples

Welcome back to another session on Engineering System Design. In the last lecture, we discussed about the functional decomposition and how to use the IDEF0 template to decompose the functions in to its sub functions.

As mentioned earlier, IDEF0 is a standard developed by us air force for development of engineering systems, and this can be effectively used for functional decomposition. It actually uses a graphical format as well as text format to represent the functions and their decompositions. And once we get these decompositions or this sub functions, we can use it for developing a functional architecture or a hierarchical function structure. We discussed about one case study that is an elevator system design and then how to use the IDEF0 diagram for decomposition of elevator system.

Today, we will discuss another example practical case study or a practical example and use the IDEF0 diagram to decompose the functions into its small functions.

(Refer Slide Time: 01:22)



Example:
Unified Data REcording System - UDARE

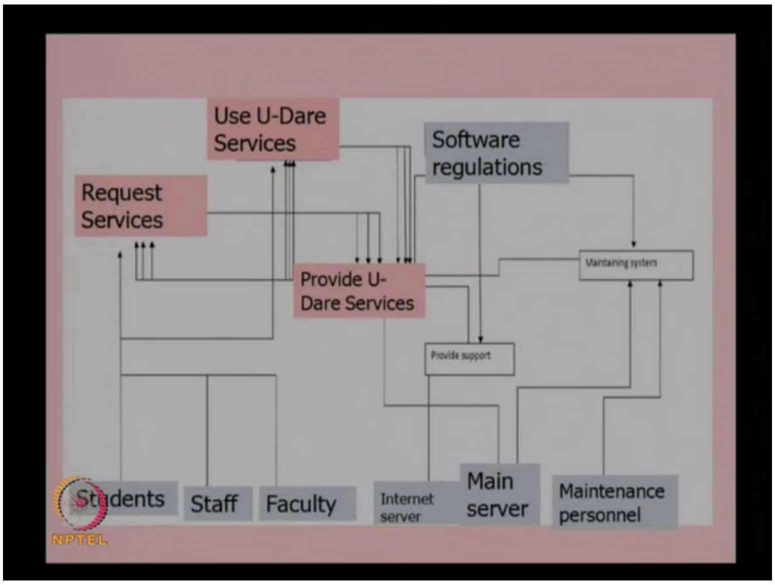
Objectives:
- Online recording and compilation of attendance for students/staff/faculty on day-to-day basis.
- Real time analysis of slot-wise engagement of students/faculty.
- Serve as a real time data base for leave/salary/scholarship computation.
- Serve as a real time data base for students' feedback on courses/ feedback analysis.
- Serve as a real time centralized data base for fees records of all students.
- A real time data base for venue allocation/time slots for year long Lit-Soc activities happening in the institute

So, the example here I am using is a unified data recording system or it is known as UDARE. Basically, this is used to record the data of educational institution especially the attendance the score the grades and all other as activities related to the a scholarship then the stripe and fees details. And apart from that it can actually act as a data a recording system as well as an information data base, whereas students as well as the staff and faculty can use it for various other purposes.

So, the main objective of these system are one is to basically online recording and compilation of attendance for students staff faculty on a day to day basis, other one is to do a real time analysis of a slot wise engagement of students and faculty basically do find out whether the student are free on a particular slot or a faculty is available on a particular slot for a meeting or for some other classes serve as a real time data base for leaves salaries scholarship, etcetera and the computation of these data and serves as a real time data base for student's feedback on courses and feedback analysis and serve as a real time a centralized data base for fees records of all students and a real time data base for when you allocation time slots for yearlong liter or social activities happening in the institutes.

So, these actually access a uniform data base and you can be used for various purposes apart from the academic activities it can be used for other cultural and social activities also. So, if you look at this system you can actually see that that actually encompasses many sub system and external system and it request a system approach in the a development of the UDARE system.

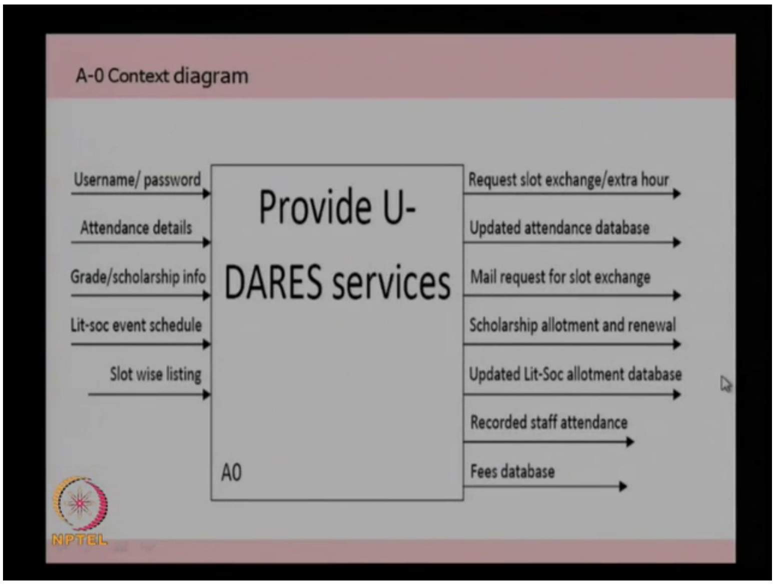
(Refer Slide Time: 03:25)



If you look at the external system diagram for this particular system, then you can see that the students, staff faculty, internet provider, main server and maintenance personnel these are the external systems directly interacting with the main system of UDARE system. And the UDARE system will provide the services which we already discussed and there are many other functions taking place the customers or the users will ask for the services and the various forms of request will be there and there are other external system like software regulations and building regulation which will be acting as a an external system for these system.

So, when we developed the UDARE system and try to decompose the function we need to clearly identify the external system separately and then look at the main function or the UDARE function only during the functional decomposition. So, these are all the external system. So, the external system will be having their own functions, but we should look only at the interactive functions or the interface function which actually will be use for interfacing the external systems to the main system.

(Refer Slide Time: 04:44)



A-0 Context diagram

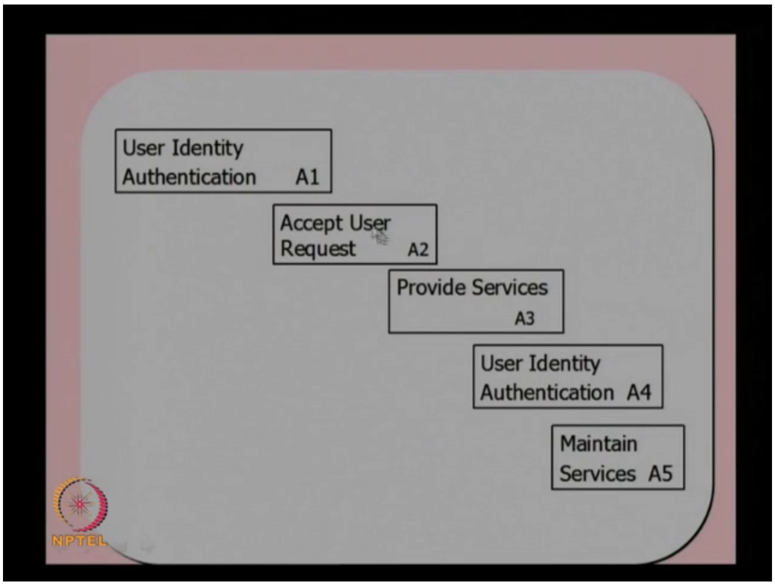
Username/ password
Attendance details
Grade/scholarship info
Lit-soc event schedule
Slot wise listing
Provide U-DARES services
A0
Request slot exchange/extra hour
Updated attendance database
Mail request for slot exchange
Scholarship allotment and renewal
Updated Lit-Soc allotment database
Recorded staff attendance
Fees database

So, this is the external system diagram for the UDARE system, if you look at the context diagram or the hyphen 0 diagram of the IDEF0, you can see that the inputs and outputs are mentioned here. So, there are many kinds of inputs about the user interface in times of password and username, then the attendance details the information regarding the grade and the scholarship and fees, then other events schedule and then the slot wise listing. Of courses, these are all coming as inputs to the UDARE system. And there are many other outputs coming from the system basically the updated attendance the mail requests for slot exchange then scholarship allotment and renewal data then updated lit soc allotment database recorded staff attendance fees database all these things are coming as output.

And you will be having many control inputs not shown if here control inputs in terms of the availability of various classrooms availability of the fees structure and the scholarship structure for different category of students like M. Tech, PhD, B. Tech. These are all the control inputs coming to the system and of course, the mechanical system will be the UDARE system or the main hardware which is used for providing the service. So, this is the context diagram from the system.

Now, this context diagram you know that A 0 is the main function provide UDARE services is the main function. Now we need to see how to decompose this into its sub functions. So, that would be coming in the first level diagram or the A 0 diagram in A 0 diagram we will try to decompose these functions into A 1, A 2, A 3 a and etcetera that is depending on the number of sub functions; we will divide this into many functions.

(Refer Slide Time: 06:35)



User Identity Authentication A1

Accept User Request A2

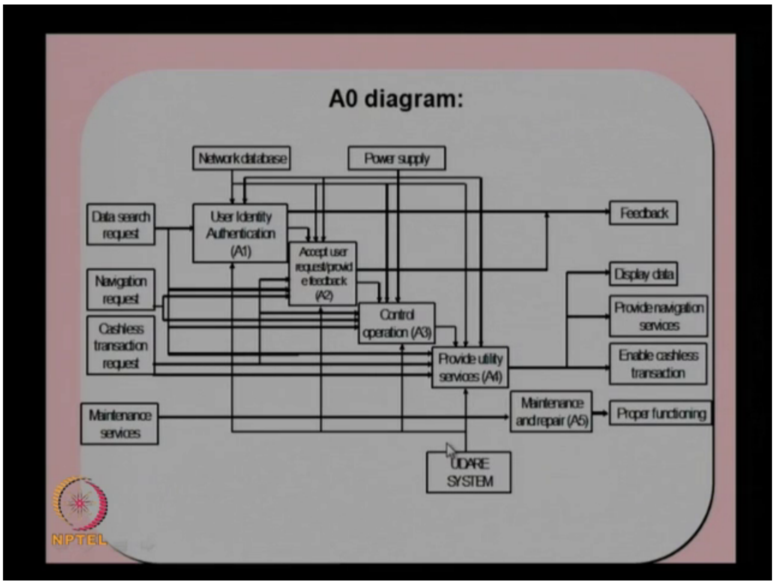
Provide Services A3

User Identity Authentication A4

Maintain Services A5

So, we can see here the main function provide UDARE services this came actually be decomposed into 4 sub functions of the main function that is user identity authentication accept user requests provide services and maintain services. So, these are the main 4 functions.

(Refer Slide Time: 06:55)



A0 diagram:

Network database

Power supply

Data search request

User Identity Authentication (A1)

Accept user request/provid e feedback (A2)

Navigation request

Cashless transaction request

Control operation (A3)

Provide utility services (A4)

Maintenance services

Maintenance and repair (A5)

UDARE SYSTEM

Feedback

Display data

Provide navigation services

Enable cashless transaction

Proper functioning

Now, if we write the A 0 diagram here user identity authentication is A 1, then accept user request provide feedback control operation provide utility services and maintenance and repairs are the 5 function which actually can be obtained from the A 0 diagram.

Now, if we look at the various inputs and controls coming into the system as well as the outputs you can see the data search request will be coming from the user and then there will be a network database which actually uses the user name and passwords and there will be other requests for coming from users like navigation requests and transaction requests. And then maintenance and all these inputs will be coming over to these functions like user identity authentication and that will be giving a feedback to the customer about whether the requests has been accepted over the password and user name is approved.

And then other requests from this data will be given to the next function which is accept user request provide feedback and again, there will be many requests coming from here the data search requests various kinds of requests should be going to this function also and there will be a feedback going from the system to the output.

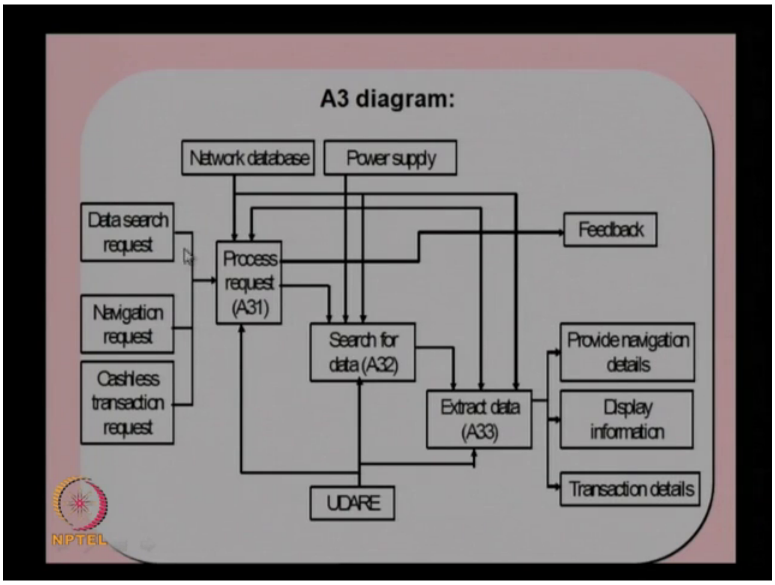
And the third function it is control operation in control operation which is the A 3 that the function of the main function. So, control operation is basically to control the use the input data the output from these 2 functions as well as the data request and then control signals will be used you have to control the operation and based on the utility services will be provided by the system. So, this is the fourth function.

And of course, maintenance and repair will be a additional function which will be actually having the output from all these functions about the malfunctioning of any of this system data these systems will be send to the maintenance and repair function. And here the any output regarding the functioning of the system will be provided and this will be given as a feed back to other systems also. So, that any malfunctioning of the system can be easily recorded as well as given as an output to the user.

So, this is the first level of decomposition or known as the A 0 diagram now if you decompose one of these functions to its sub function we will be getting the A 1 diagram in A 1 diagram. We will actually sorry; this is the detailed explanation for the previous diagram as you can see here this is the user authentication A 1 and this is the A 2 function and this is the A 3 function and this is the A 4 function and then 5 function and the all the output from this block are shown here like provide utility services of various utility like data display navigation services cashless transaction all those output will be provided by this particular function.

First level diagram in the first level diagram, we will be decomposing the A 3 function into its sub functions. So, you can see here. So, A 3 function is basically to provide the control operation. So, in control operation we need to control the request from the users and then basically based on the request we need to provide outputs.

(Refer Slide Time: 10:19)



A3 diagram:

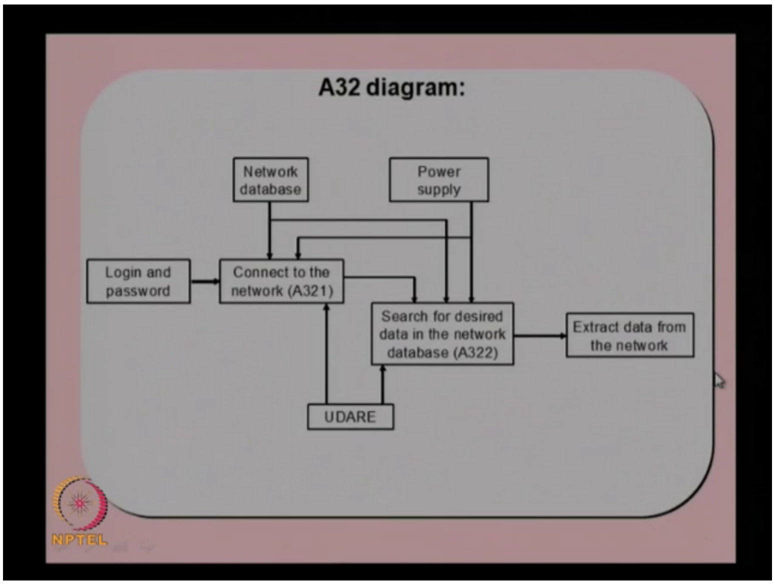
So, here you can see this A 3 function A 3 has got many sub functions basically process request which actually process the request received from the customers or the users and then search for data based on the request the data will be searched for the requested information and then it will be extracting the data and then be providing it to the provide service function.

So, these are the 3 sub functions of the A 3 function and that is the level 1 diagram. So, here you can see the data search request navigation requests and cashless transaction requests all kinds of requests from the users which will be coming from the in terms of in digitized format to this function and here this will be a processed and the processing information based on the processing information a feedback will be given whether the data is possible to provide the service or not or is there any a malfunctioning of the system. And then, that information the process information will be sent to the database for searching for the data and whatever the data is obtained will be given to the extracted the data and will be given as an output to the next level. So, A 4 function or to the output.

So, this is the level 1 diagram and as you can see the network database will be used as an input here and the other input request should be also used of course, power supply and other control inputs also will be coming to this in order to provide the service again further dividing this function if you want to go for further details and to see what are the other functions needed to provide this particular A 1 of these function. So, we can take A 32 as a function and then try to decompose that into further level this again detailed description of the previous diagram A 3 diagram, if you can see A 31 functions, here A 32 function and then A 33 function. So, these are 3 sub functions of A 3 and their control and interaction are shown in this diagram.

So, you can see the various feedback and outputs provide navigation details display information and transaction details all are shown over here now if you decompose this function further A 32.

(Refer Slide Time: 12:39)



A32 diagram:

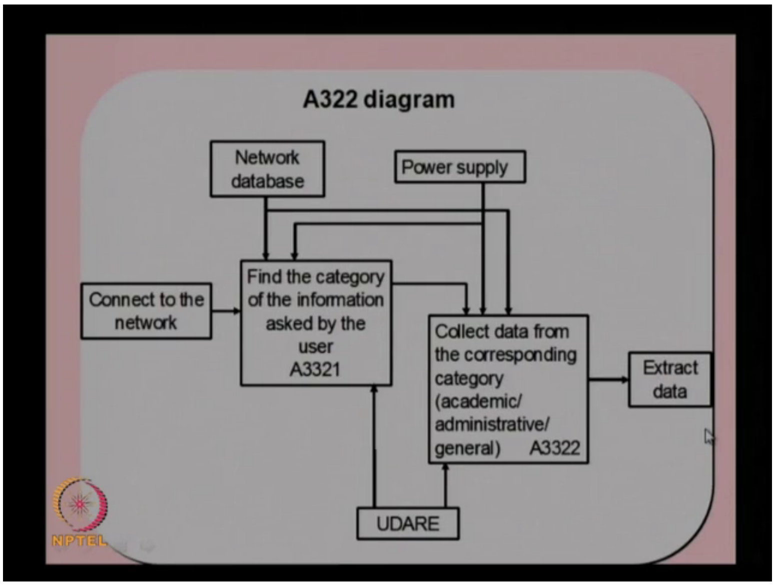
Network database
Power supply
Login and password
Connect to the network (A321)
Search for desired data in the network database (A322)
Extract data from the network
UDARE

And you will be getting the 3 2 diagram which is level 2 diagram in level 2 diagram, we will try to decompose the function A 32 to its subcomponents. So, you can see that here A 321 and A 322 are the 2 sub functions of A 32.

So, in A 3 to connect to the network is basically to search for data. So, connect to the network is one function and then search for desired data the network database is another function. So, these 2 functions will provide you the data which is needed to provide the service. So, the extract data from the network will be the output from this block again the login and password information will be provided in order to search the network database and then the desired data will be given to the next level function.

So, this is how we decompose the function A 32. Now if you want to decompose this further we can actually decompose this into the next level. So, 3 to 1 or 3 to 2 can be decomposed into level 3 diagram and the this actually shows the detail explanation of the previous one A 32 diagram. So, connect to the network is A 321 and the search for desired data and the network is A 322 diagram and the output is extract data from the network.

(Refer Slide Time: 13:58)



A322 diagram

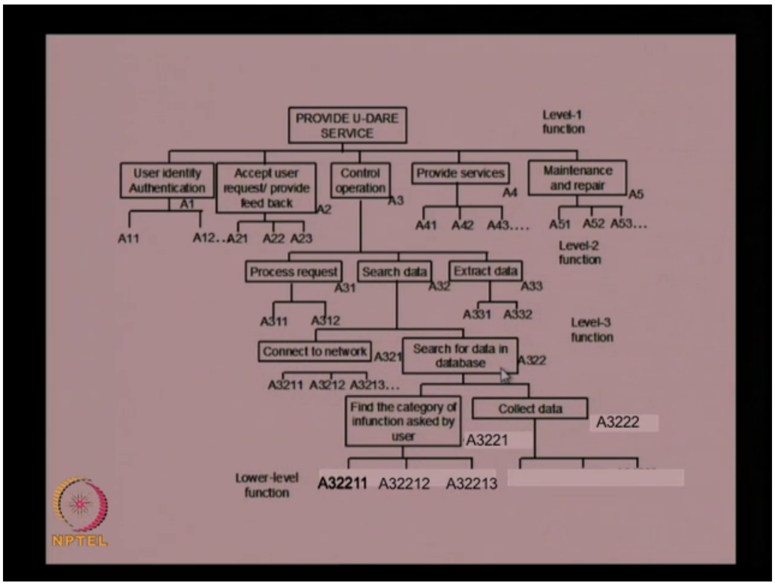
Network database
Power supply
Connect to the network
Find the category of the information asked by the user A3321
Collect data from the corresponding category (academic/ administrative/ general) A3322
Extract data
UDARE
NPTEL

So, we will go to the next level of decomposition where A 322 is decomposed into its sub functions. So, that is known as A 322 diagram and this is a level 3 diagram and in this case the A 3 to 2 function is decomposed into 3321 and 3322. So, here the 332 function the first subdivision is find the category of the information asked by the user and then the second is collect data from the corresponding category. So, it depending on the category the data will be collected from the corresponding category.

So, we can see that A 322 function has for 2 sub functions and the main output is coming from the a function A 3322 which is the data extracted. So, like this if you go by level 0, level 1, level 2 and level 3 diagrams we can actually keep on decomposing this function into its sub functions and to what level we need to do this is actually determined by the type of function, but in general we go up to level 3 or level 4 functions and by that time we will be having a function which need not be decomposed further or a module or hardware form in the form of a hardware or in terms of a software will be available to provide this function. So, there we actually stop the decomposition like collect data from the corresponding category.

So, this actually can be usually implemented using a software code. So, we do not decompose this into further because it is easily understandable at this level, in case it is not so easy to understand or if you feel that there is a need for further decomposition we can go to the level 4 diagram or level 5 diagram. And, that actually depends on the situation and the understanding of the function by the person who actually develops the functional decomposition. So, this is how we actually use the IDEF0 diagram to decompose the main function into its sub functions this again shows the same previous diagram in a more clarity that A 3321 function and A 3322 function are shown here in this diagram.

(Refer Slide Time: 16:11)



So, once we have this decomposition basically what we did was to take the main function and then see what are the sub functions needed for this those functions are identified through the decomposition through IDEF0 diagram and once we have these sub functions we need to represent it in a hierarchical format what is the top level function what are the level 0 function, then level 1 function, level 2 function like that and that is represented in a hierarchical way then we will get the hierarchical functional structure and that is what is shown here.

. So, you can see that level 0 is provide UDARE services. So, this is the function for providing the main function that is the UDARE services. So, that is the level 0 diagram then this level 0 diagram was decomposed into 5 functions basically user identity authentication accept user request provide feedback control operation provide services and maintenance and repair. So, you can see that A 1, A 2, A 3, A 4, A 5 are the level 1 functions and then these level 1 functions each of these functions can be decomposed into its sub functions like A 11, A 12 or A 21, A 22, A 23, etcetera A 31, A 32, A 41 like this it can be decompose.

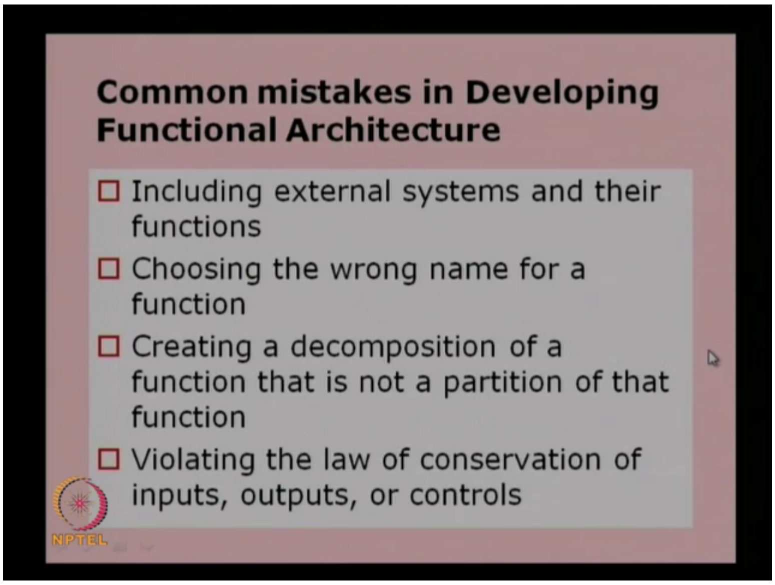
This case study we actually decompose the function control operation A 3 into its sub functions like A 31 A 32 and A 33 where process request search data and extract data are the sub functions again this each of these sub functions like A 31 can be decomposed to A 311, A 312, A 32 can be decomposed similarly A 33 can also be decomposed into sub functions again we can use the IDEF0 diagram to do this decomposition.

In this A 32 function we decomposed and then we found that A 321 and A 322 are the 2 sub functions basically connect to network search for data in the database. So, these are the 2 sub functions again the function A 322 was decomposed to A 3221 and 3222 and similarly, this can be subdivided if needed we can go for lower level functions and decomposed to further.

So, this actually shows a hierarchical function structure now if you want to provide this particular function, we can identify what are the lowest level functions to be provided in order to provide this function and this actually helps us to identify the proper hardware in order to provide this function. So, this is the basic way of decomposing the functions into its small functions and showing them in a hierarchical way and this is known as a hierarchical function structure for a system.

. So, this actually shows more clear view of the function. So, the provide UDARE services is the main function and these are the sub functions A 1, A 2, then A 3, then A 4 and 5 and then you can decompose this A 3 function into its sub functions A 31, A 32, A 33 and then further down you go for A 3 can be decomposed into A 32 can be decomposed to 3 2 1 3 2 2 and so on. So, we will be able to decompose all these functions into its smallest level functions and provide the hierarchical function structure. So, that is how we do the decomposition.

(Refer Slide Time: 19:59)



Common mistakes in Developing Functional Architecture

- □ Including external systems and their functions
- □ Choosing the wrong name for a function
- □ Creating a decomposition of a function that is not a partition of that function
- □ Violating the law of conservation of inputs, outputs, or controls

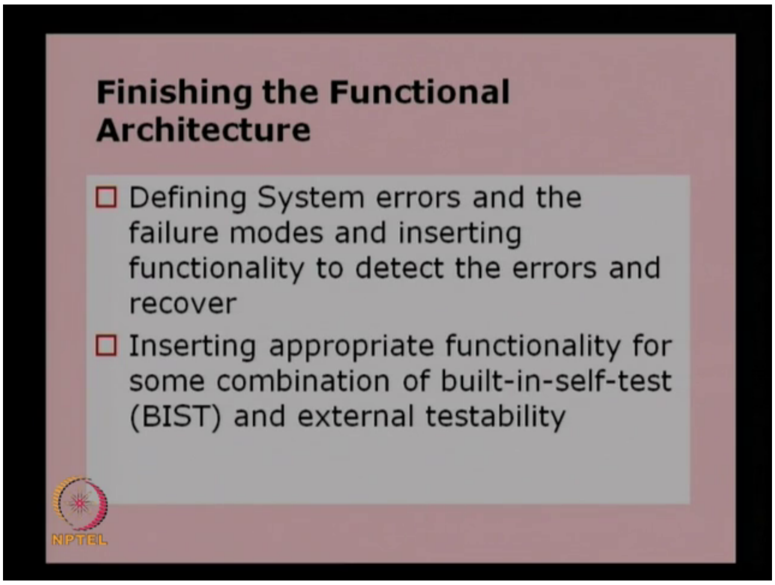
Some of the common mistakes in developing the functional architecture are we actually have different format or different methods to do the decomposition basically we discussed about hatley-pirbhai template and IDEF0 diagram and then both these methods are some common mistakes normally happen because that is students or the engineers they try to improve the external systems and their functions as part of the main function. And this is one of the reasons we defined the external system in the beginning itself we clearly state that the external systems are outside the boundary of the main system. And therefore, no external system function should be including the function decomposition.

Similarly choosing the wrong name for a function; so always the function need to be expressed in terms of a and now and ever and if you use it in a different way, then the actual the meaning of the function will not be clearly understood and therefore, we make mistakes in the further decomposition therefore, we had to choose the correct name for the function and then creating a decomposition of a function that is not a partition of that function, again we cannot have a function which is already a main function as a sub function again, therefore, when we decompose the function we should be very careful and we should not do a function which is not a part of the main function.

So, make sure that all the functions are really part of the function being decomposed and then violating the law of conservation of inputs outputs or controls basically as you can see is an every function, there is an input there is an output and some control functions. So, all the inputs should actually come out of the system as an output in form of suitably processing using the control. So, the all the signal is which actually goes in and the out there should be a balance and there should not be something which is completely absorbed in the function. So, that is actually against the law of conservation of inputs outputs or controls.

So, these points need to be taken into account while decomposing the function and make sure that we do not make the common mistakes this function decomposition is very prone to have. So, follow these rules and then try to avoid the common mistakes that are about getting the functions basically from the main function of the system, but apart from these functions we need to have some additional functions in the functional architecture.

(Refer Slide Time: 22:31)



Finishing the Functional Architecture

□ Defining System errors and the failure modes and inserting functionality to detect the errors and recover
□ Inserting appropriate functionality for some combination of built-in-self-test (BIST) and external testability

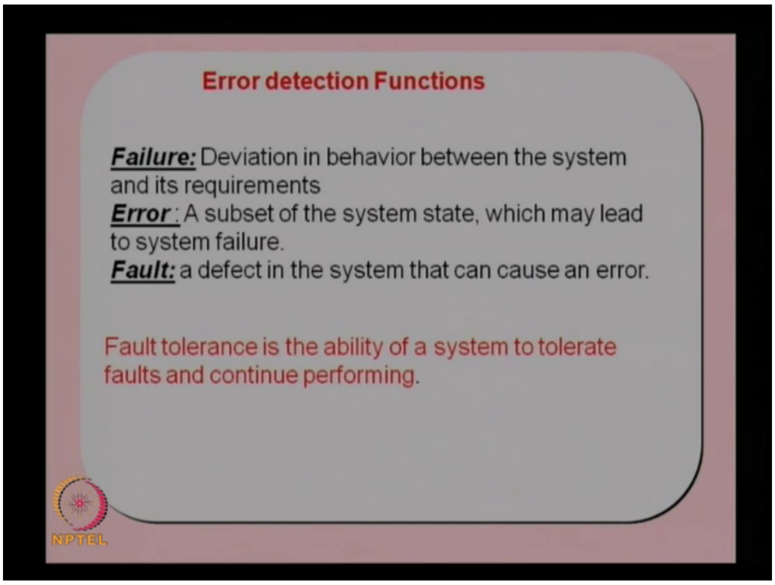
So, in order to finish the functional architecture we need to look at few more additional functions which are not very obvious to the designer in the beginning, because we always look at the user requests and then processing the request and providing the output.

So, in order to make this happen we need to provide additional functions also in the system and here what we need to look at this basically the system errors and then identification of the errors in the system. So, in order to finish the functional architecture we need to define the system errors and the failure modes and inserting functionality to detect the errors and recover. So, apart from providing the services we need to have the functions in the system in order to make sure that the system is working perfectly well and as well as it actually identifies some errors in the system or faults in the system and it is reported. So, the functionalities are needed to identify these faults that are the fault identification functions.

And the other one is inserting appropriate functionality for some combination of built in selfness and external testability. So, there should be some tests going on within the system which is known as built in test to make sure that the system is performing well throughout its life as well as external testability is possible that it is possible to test the system some external resources or external means. So, the functions to be provided in order to provide these functions also. So, if you add with these functions along with the previous functional architecture then only the functional architecture is complete.

How do we provide these? You know to provide this function we need to define some of the terms which actually defines the identification.

(Refer Slide Time: 24:21)



Error detection Functions

Failure: Deviation in behavior between the system and its requirements
Error : A subset of the system state, which may lead to system failure.
Fault: a defect in the system that can cause an error.

Fault tolerance is the ability of a system to tolerate faults and continue performing.

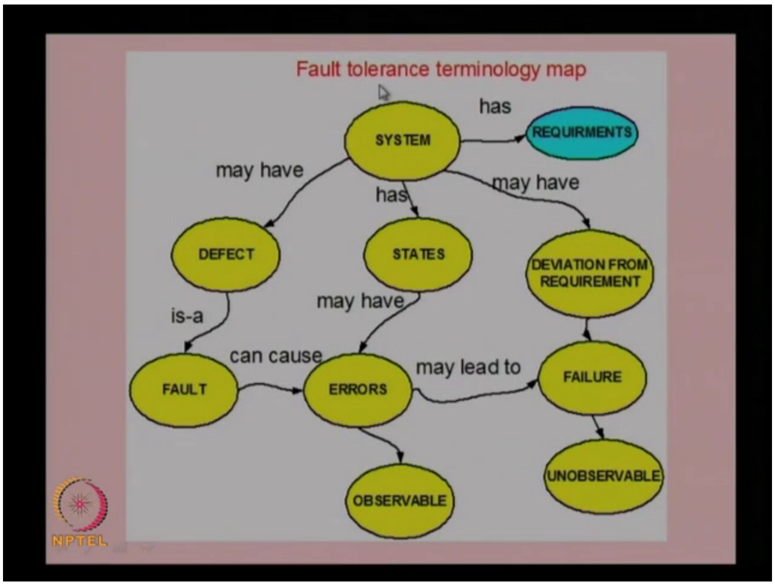
So, we define these functions as error detection functions. So, these are the functions to be provided there are the error detection functions. So, you know to find out the function or how to provide this function we need to define these terms like a failure the failure in a system is defined as a deviation in behavior between the system and its requirements.

So, every system has got some requirement identified by the users or the stakeholders. So, whenever we are not able to provide a; this requirement then it is known as a failure of the system. So, any deviation in behavior within the system and its requirements are known as a failure an error is a subset of the system state which may lead to the system failure. So, every system has got a system state in we define the system state in all of the earlier lectures basically it is a snapshot of the system characteristics at a particular time. So, subsets of the system state is errors which lead to a system failure.

Or any changes in the system state or which is which is not as per the required state, then this may lead to a system failure and subset of the state is known as an error and the fault is a defect in the system that can cause an error. So, any fault in the system in terms of its physical behavior or a malfunctioning of a physical component this can actually cause a defect and that will cause an error in the system. So, these are the term which actually comes into play when we define the error detection functions.

So, the basic idea of providing the error detection function is to provide fault tolerance in the system. So, fault tolerance is the ability of a system to tolerate faults and continue performing. So, every system should be able to tolerate the fault in the system and still continue performing. So, even when there is a small error and it actually cause results into some problem the system should be capable of continuing the performance without affecting its outputs or the performance requirements and that is known as the fault tolerance in a system.

(Refer Slide Time: 26:38)



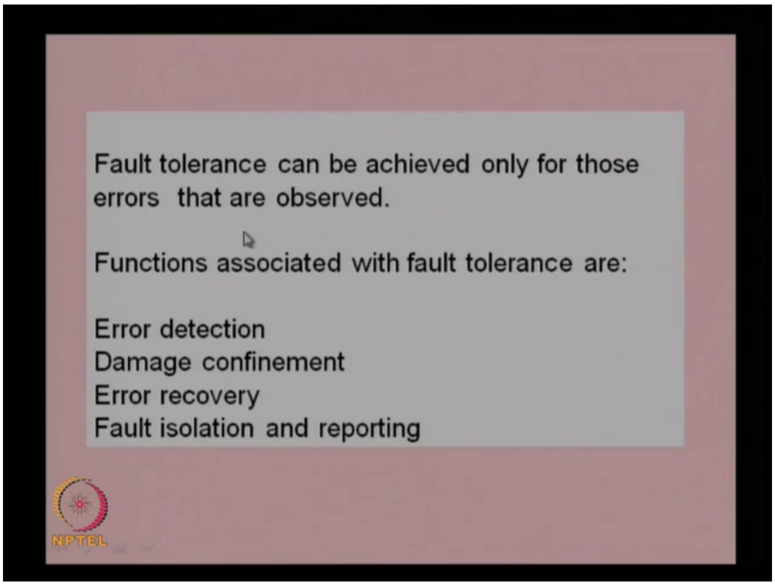
Fault tolerance terminology map

If you look at the fault tolerance terminology may have to shown here we can see that there are observable and unobservable failures the system or the errors in the system. So, every system has some requirements. And these requirements are the basic objective or output of the function or the system and then system may have deviation from the requirement. So, there are requirements for the system and there may be a deviation from the requirement and this is known as the failure of the system and the system as such cannot observe the failure. So, because the system does not know what is it supposed to do only the customer will be knowing or the consumer will be knowing what it has to do.

So, the system as such cannot identify its failures. So, that is why is are a unobservable failures, but we need to convert this into observable count it is then only we will be able to correct or have fault tolerance in the system. So, we can see a system has many states and states may have errors and this errors are observable these are the errors which actually lead to failure. So, we look at the system states and then these errors and since these errors are observable; we look for observable erase and then try to observe the error and then provide a fault identification or fault tolerance function.

Again the system may have defect which is a fault and this can actually cause errors; so the basic culprit here maybe a defect or lack of providing the requirement. So, any one of this can actually be observed through the errors and then these errors can be monitored and necessary functions can be provided in the system in order to provide the fault tolerance and these functions which actually provide these fault orders are known as for tolerant functions.

(Refer Slide Time: 28:30)



Fault tolerance can be achieved only for those errors that are observed.

Functions associated with fault tolerance are:

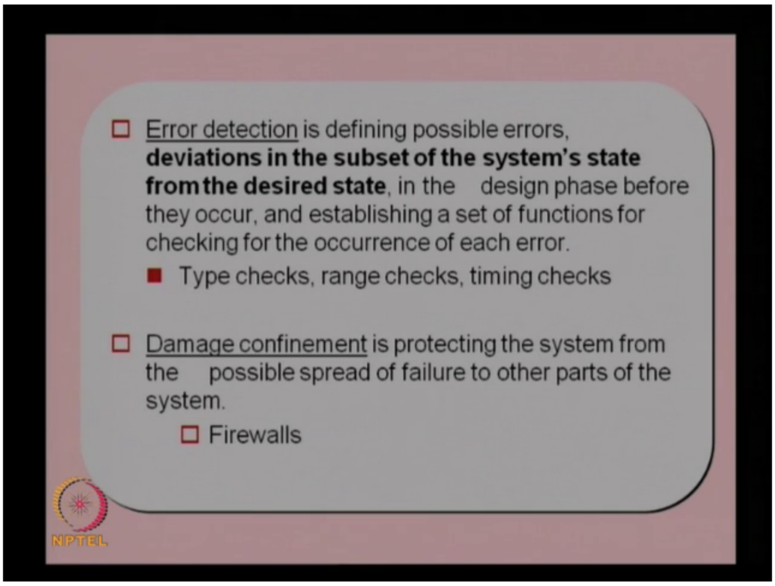
Error detection
Damage confinement
Error recovery
Fault isolation and reporting

So, fault tolerance can be achieved for those errors that are observed. So, there are many functions in order to provide the fault tolerance basically if you look at the fault tolerance function there should be an error detection function basically to detect an error if there is an error or not or when something goes wrong the system should be capable of identifying the error and then only we can actually tolerate that particular error and then there will be a damage confinement basically when there is a damage we need to make sure that it is not spreading to other areas. So, this is known as damage confinement and then error recovery.

So, if there are a error happened how do we recover from that particular error that is known as error recovery function and then fault isolation and reporting. So, we can see here fault isolation and reporting is the final one basically you isolate that particular fault and report to the maintenance function or the maintenance module so that we can take the corrective action. So, when all these 4 functions are provided and you have the error detection damaged confinement error recovery and fault isolation reporting then we are getting the fault tolerance in the system.

And this can actually be provided by different ways different functions can be incorporated into the system in order to have all these 4 functions. And, when we add these functions into the function architecture then the fault tolerance function may becomes possible in this particular system.

(Refer Slide Time: 30:03)



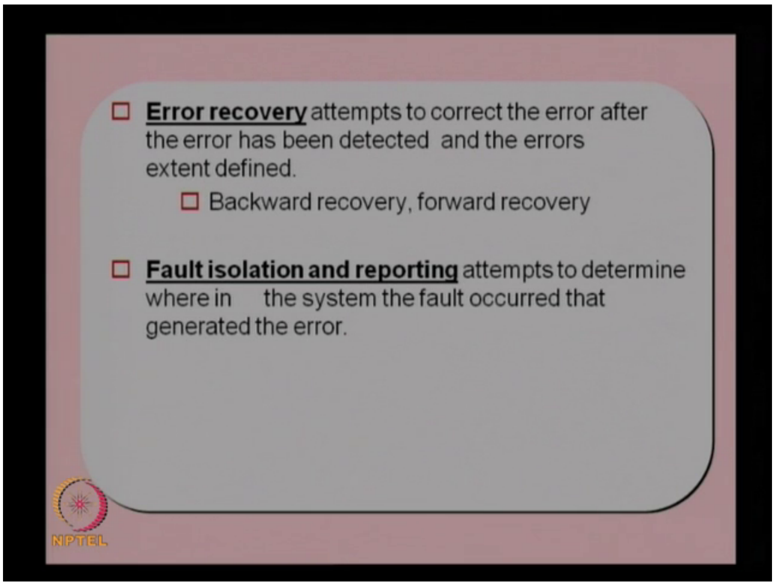
- □ Error detection is defining possible errors, **deviations in the subset of the system's state from the desired state**, in the design phase before they occur, and establishing a set of functions for checking for the occurrence of each error.
    - ■ Type checks, range checks, timing checks
- □ Damage confinement is protecting the system from the possible spread of failure to other parts of the system.
    - □ Firewalls

So, error detection as I told you the first function is error detection this is defined as error detection is defining possible errors or the deviations in the subset of the system state from the desired state as we know that an error is basically a variation from the system state from the desired state. So, we actually defined this using the error detection function in the design phase before they occur and establishing a set of functions for checking for the occurrence of each error. So, we know the state and we know; what are the states to be observed and we define some functions or a set of functions for checking the occurrence of each of this error.

Normally the functions are type checks range checks and timing checks type checks is the type of data whether it is an integer data or a floating data or its some other kind of at a digital data or an analog data or there range checks. So, what is the range of data expected from that particular output and if it is crossing the range then an error will be detected or an error will be reported similarly timing checks. So, if you do not get a particular output from source in a particular duration then error will be declared.

So, thus these are the different types of functions provided in the system to have error detection type checks range checks and timing checks and damage confinement is protecting the system from the possibility spread or failure to other parts of the system. So, if there is a failure and this failure should be protected or failure should be prevented from spreading to other parts of the system and this is known as damage confinement normally firewalls are provided in order to prevent this happening. So, what are the damage it will be confined to a particular region only and they do not spread to other areas.

(Refer Slide Time: 32:04)



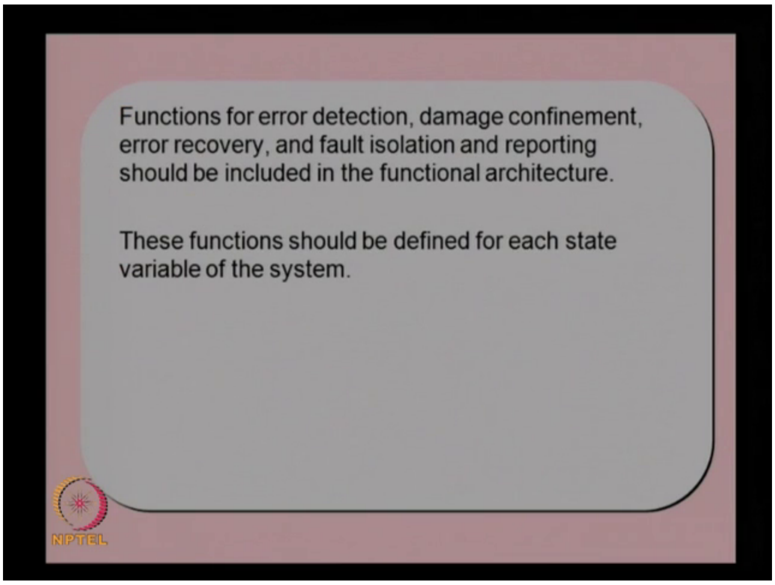
- □ **Error recovery** attempts to correct the error after the error has been detected and the errors extent defined.
  - □ Backward recovery, forward recovery
- □ **Fault isolation and reporting** attempts to determine where in the system the fault occurred that generated the error.

So, that is damage confinement the other one is error recovery error recovery attempts to correct the error after the error has been detected and the errors extent defined. So, once the error has been detected and its extent defined, then it will start recovering from this error. So, there are many ways of doing it backward recovery or forward recovery backward recovery is basically will go back to its previous state where there was no error and then try to take that data. And then, go ahead with that data and continue its calculation or continue the processing. So, that is known as backward recovery in forward recovery it will try to see the next correct data and from that point onwards it will take the data and then start processing. So, this is known as forward recovery. So, any one of these can be employed depending on the system to provide an error recovery function.

When the last one is fault isolation and reporting it is actually attempts to determine where in the system the fault occurred and that generated the error. So, this is basically to find out the source of error. So, it will try to determine the point where or the place where they actually error occurred and that will be reported. So, that is the fault isolation and reporting and there are many ways of doing this you can do it using a software or you can use this hardware and the next chapter we will discuss about how to provide these functions using hardware or what kind of hardware can be used for fault identification or add isolation reporting as well as for other functions of error recovery and error reporting as well as prevention of the error work from spreading.

So, all this can be done using hardware or software. So, these are the main functions to be provided in order to have a fault tolerant system.

(Refer Slide Time: 33:51)

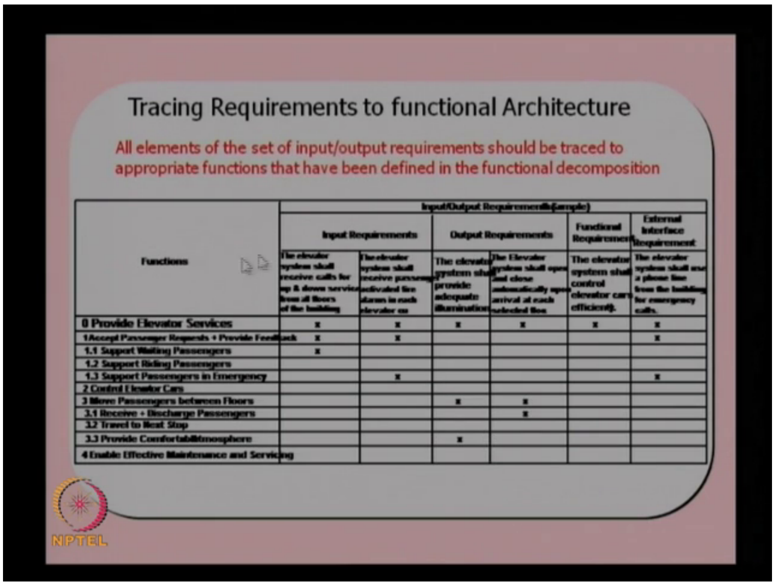


Functions for error detection, damage confinement, error recovery, and fault isolation and reporting should be included in the functional architecture.

These functions should be defined for each state variable of the system.

So, the functions for error detection damage confinement error recovery and fault isolation and reporting should be included in the functional architecture and this function should be defined for each state variable of the system. So, depending on the number of state variable we will be defining the functions for all these 4 the these error detection damage confine confinement error recovery and fault isolation and reporting for each state and make sure that it is possible to have a fault tolerant system for all these important states of the system.

(Refer Slide Time: 34:26)



Tracing Requirements to functional Architecture

All elements of the set of input/output requirements should be traced to appropriate functions that have been defined in the functional decomposition

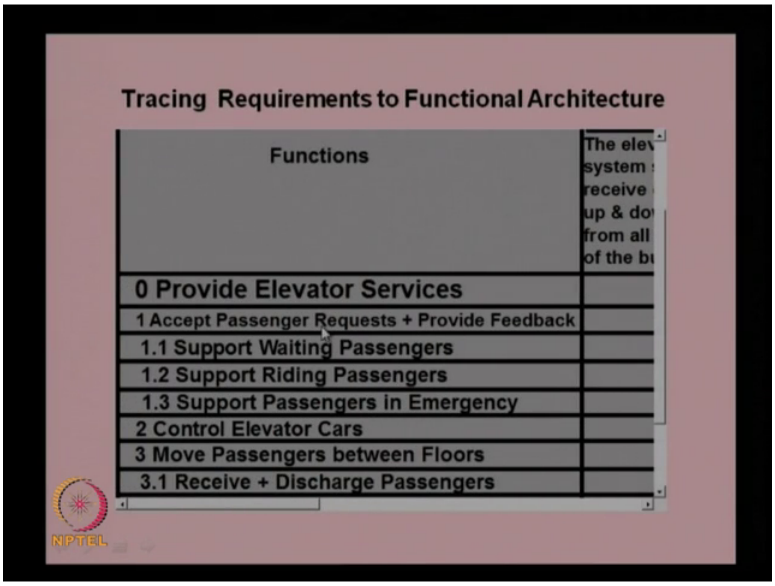
Functions	Input/Output Requirements (Example)					
	Input Requirements		Output Requirements		Functional Requirement	External Interface Requirement
	The elevator system shall receive calls for up & down service from all floors of the building	The elevator system shall receive passenger activated fire alarms in each elevator ca	The elevator system shall provide adequate illumination	The Elevator system shall open and close automatically upon arrival at each selected floo	The elevator system shall control elevator cars efficiently.	The elevator system shall use a phone line from the building for emergency calls.
0 Provide Elevator Services	x	x	x	x	x	x
1 Accept Passenger Requests + Provide Feedback	x	x				x
1.1 Support Waiting Passengers	x					
1.2 Support Riding Passengers						
1.3 Support Passengers in Emergency		x				x
2 Control Elevator Cars						
3 Move Passengers between Floors			x	x		
3.1 Receive + Discharge Passengers				x		
3.2 Travel to Next Stop						
3.3 Provide Comfortable Atmosphere			x			
4 Enable Effective Maintenance and Servicing						

So, that was about developing the function architecture using methods of HP template and IDEF0 as well as including the functions using fault tolerant system. So, we can actually incorporate all these functions in the system and identify all these function in the system and provide a functional architecture. So, apart from identifying the function this function architecture can be used for identifying the requirements or tracing the requirements to the functional architecture we know that there are different requirements and we have provided different functions.

Now, we need to may ensure that these functions are sufficient to provide the requirements or we can actually trace the functions or the requirements and then associate the corresponding function to these requirements. So, elements of the set of input output requirements should be traced to appropriate functions that have been defined in the functional decomposition. So, when we have these input output requirements we need to find out the; which function is basically satisfying this particular requirement or which are the functions which actually can be used for satisfying the particular requirements. So, that can be obtained using the functional architecture or we can use the functional architecture to trace these requirements or to find out the relation between the requirements as well as the functions provided in the system.

So, for this will use a chart which is basically show the functions here and the requirements I will show you a more detailed diagram. So, you can see here this is how we actually trace the requirements.

(Refer Slide Time: 36:03)



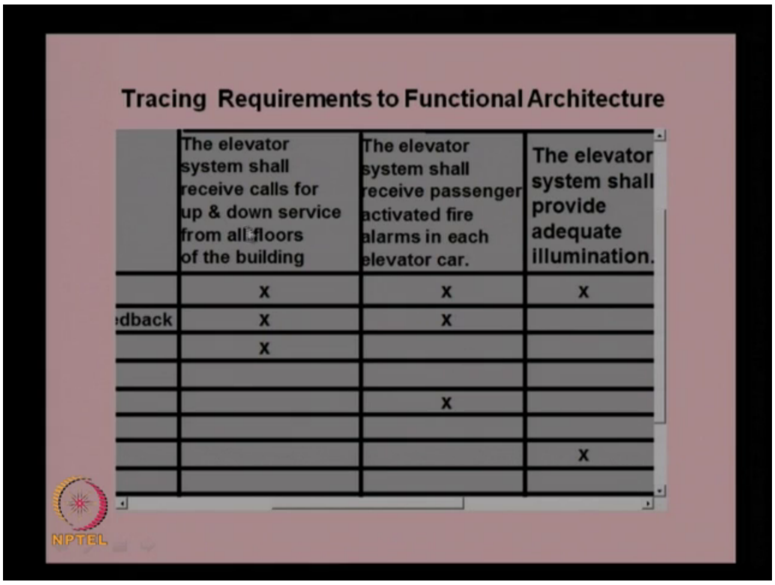
Tracing Requirements to Functional Architecture

Functions	The elev system receive up & do from all of the b
0 Provide Elevator Services	
1 Accept Passenger Requests + Provide Feedback	
1.1 Support Waiting Passengers	
1.2 Support Riding Passengers	
1.3 Support Passengers in Emergency	
2 Control Elevator Cars	
3 Move Passengers between Floors	
3.1 Receive + Discharge Passengers	

So, we write down the requirements here. So, suppose you take the provide elevator service as one requirement accept passenger request and provide feedback will be main one function of this then support waiting passengers support riding passenger support emergency passengers; passengers in emergency and control these are the sub function.

We need to sub requirements basically these are the requirement from the customer. So, we need to find out which are the functions actually satisfy these requirements.

(Refer Slide Time: 36:41)

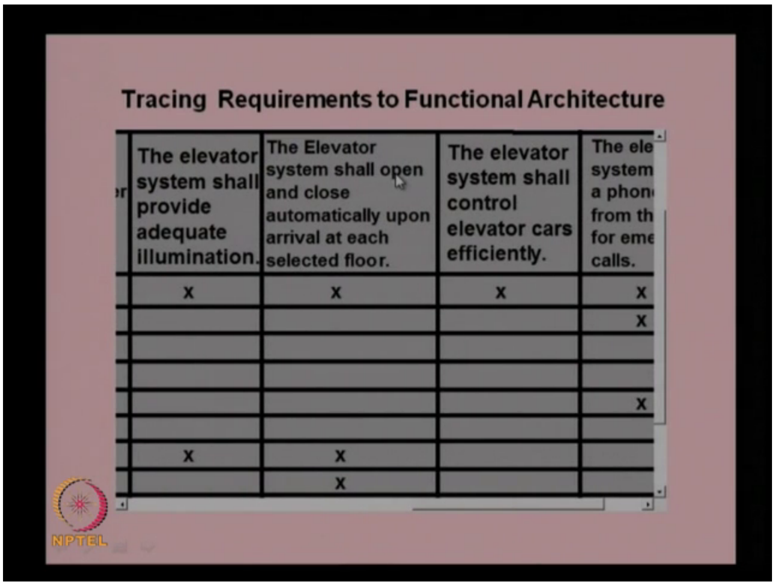


Tracing Requirements to Functional Architecture

	The elevator system shall receive calls for up & down service from all floors of the building	The elevator system shall receive passenger activated fire alarms in each elevator car.	The elevator system shall provide adequate illumination.
	x	x	x
dback	x	x	
	x		
		x	
			x

So, if you look at the provide elevator services we can see that there are many functions the elevator system shall receive calls for up and down service from all doors of the building, then the elevator system shall receive passenger activated fire alarms in each elevator car elevator system shall provide adequate illumination.

(Refer Slide Time: 36:57)



Tracing Requirements to Functional Architecture

r	The elevator system shall provide adequate illumination.	The Elevator system shall open and close automatically upon arrival at each selected floor.	The elevator system shall control elevator cars efficiently.	The ele system a phon from th for eme calls.
	x	x	x	x
				x
				x
	x	x		
		x		

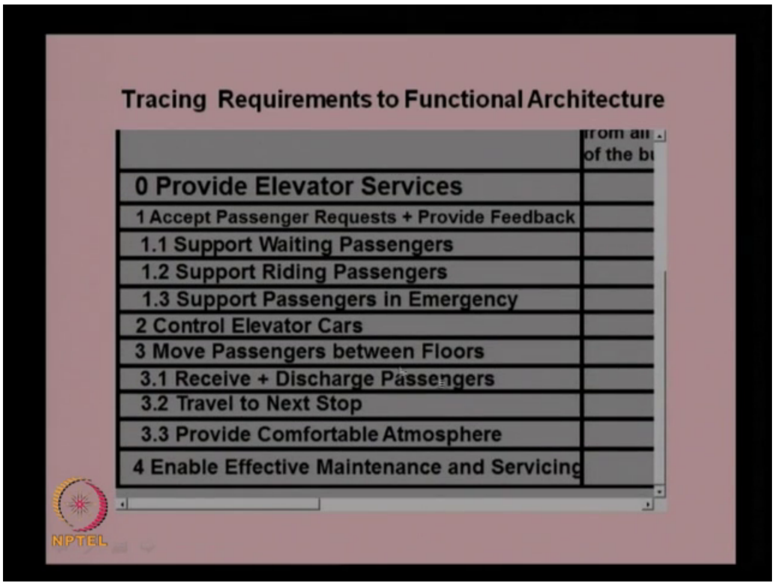
These are the requirements of the customers the elevator system shall open and close automatically upon arrival at each selected floor.

The elevator system shall controlled elevator cost efficiently and then elevator system shall use a phone line from the building for emergency calls.

So, these are all the requirement from the customer we need to see how these requirements are satisfied by providing the functions. So, this actually support waiting passengers support riding passengers and support passengers in emergency these are the 3 sub functions we provided and we can see that the elevator system shall receive calls for up and down service from all floors are satisfied using these 3 functions. So, these 2 functions provide elevator services function satisfy the requirement of elevator systems shall receive calls this again with satisfies this requirement also system shall receive passenger activated fire alarms for an emergency case similarly system shall provide adequate elimination.

So, all these functions if you look at these functions you can see that all these functions are used for satisfying the provided elevator service requirement.

(Refer Slide Time: 38:07)



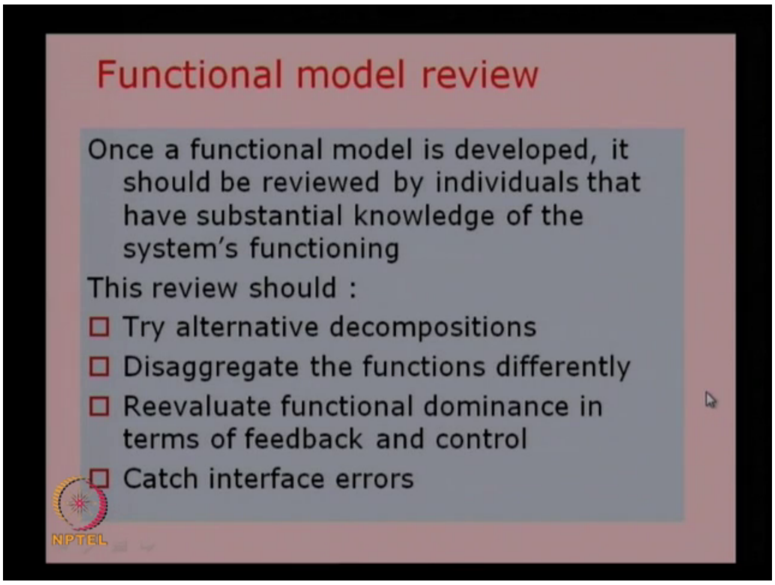
Tracing Requirements to Functional Architecture

	rom all of the bu
0 Provide Elevator Services	
1 Accept Passenger Requests + Provide Feedback	
1.1 Support Waiting Passengers	
1.2 Support Riding Passengers	
1.3 Support Passengers in Emergency	
2 Control Elevator Cars	
3 Move Passengers between Floors	
3.1 Receive + Discharge Passengers	
3.2 Travel to Next Stop	
3.3 Provide Comfortable Atmosphere	
4 Enable Effective Maintenance and Servicing	

Similarly, we can actually find out the other requirements like control elevator cars or move passenger between floors we can identify which are the functions. These are the requirements satisfied by this function or which are the function basically satisfying this requirement you can look at this the elevator system shall provide adequate illumination is one requirement we can see that there are 1 2 1 2 and 3 3 functions are there which actually satisfy this requirements.

. So, we can have this kind of a traceability of the requirements and the functions or we can identify the each; we can take each requirement and then see; what are the functions which actually satisfy this requirement. So, apart from developing the functional architecture and identify the functions we can actually be used for tracing the requirements also. So, this only a sample we can actually take any particular requirement and then see how their particular requirement is satisfied through the various functions. So, this is another advantage or another use of having the function architecture.

(Refer Slide Time: 39:12)



Functional model review

Once a functional model is developed, it should be reviewed by individuals that have substantial knowledge of the system's functioning

This review should :

□ Try alternative decompositions

□ Disaggregate the functions differently

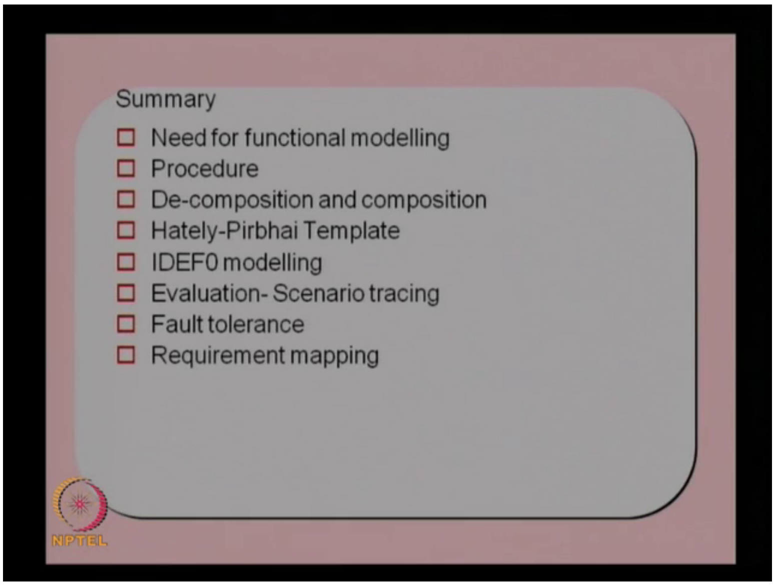
□ Reevaluate functional dominance in terms of feedback and control

□ Catch interface errors

So, once we have this functional model or the functional architecture developed we need to reviewed by we test be reviewed by another position or another individual who actually has got substantial knowledge of the systems functioning. So, whenever we develop their functional architecture there is a chance that we; some of these points or there was an error unknown error or some kind of discrepancy in the function decomposition. So, this needs to be unless once we have the preliminary model of the function structure.

So, the review should try alternative decompositions. So, we need to see whether we get decomposed a function by a different method or desegregate the functions in a different way and reevaluate functional dominance in terms of feedback and control. So, we have some control inputs and then there may be some feedback coming from some functions. So, we have to reevaluate the function dominance in terms of feedback and control also similarly catch the interface errors or whenever we have different functions where there I mean whenever there is an interface between functions we need to look at the errors which may come up within these interfaces. So, all these can be done in order to review the functional model and based on this model we can actually refine the functional structure also.

(Refer Slide Time: 40:30)



Summary

- □ Need for functional modelling
- □ Procedure
- □ De-composition and composition
- □ Hately-Pirbhai Template
- □ IDEF0 modelling
- □ Evaluation- Scenario tracing
- □ Fault tolerance
- □ Requirement mapping

So, to summarize whatever we discussed in the last 3 lectures, basically we started with the need for functional modeling, we mentioned about the previous lectures about how do we identify the requirements. And once we identify the requirements it is the next logical step is basically to convert these requirements into functional structure that is to identify the required functions it will satisfy these requirements.

So, that is where we started for the functional modeling to make sure that we can satisfy the requirements through some functions and then we discussed about procedure for doing the decomposition and we mentioned that there are many methods like we go for a top down approach or a bottom approach to decompose and then we may discussed about the 2 methods one is known as the hatley-pirbhai template or its known as hp template the other one is the IDEF0 modeling and these 2 methods can be used for top down decomposition of the functions and then we saw few examples and discussed how do we use these methods to decompose the functions.

And we identified some of the important additional advantage of using this function decomposition basically one was to evaluate the scenario tracing to take a particular scenario and then identify all the functions UDARE for that particular scenario and then see whether there is anything missing or there is any overlap or redundancy in the function. And we discussed about the fault tolerance functions how do we actually complete the functional architecture where providing by providing the fault tolerance functions we discussed about error detection functions error confinement functions and the error reporting functions and soon. So, these functions also need to be incorporated into the fault tolerance and the we discussed about the requirement mapping basically how do we map the requirements to the functions using the functional architecture.

So, these are the points we discussed in the last few lectures. In the next lecture we will be basically looking at the possibility of how do we actually convert these functions into physical architecture. Basically, these functions are needed or satisfy the requirements, and once we identify this function the next task is basically to how do we get a physical architecture for the system which will satisfy the requirements. So, the physical architecture development is the next logical step of system development. And, we will discuss about this in the next few lectures, till then goodbye.