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

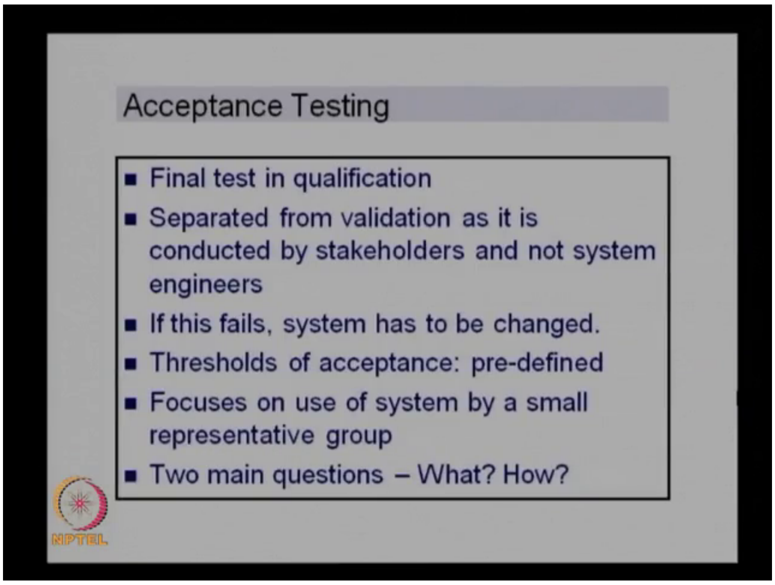
Lecture - 20
System Design Example: AutoLink System

Dear friends, welcome back this is the another session on Engineering System Design. Last two lectures we have been discussing about the integration and qualification of engineering system which is the last stage in the design of engineering systems. So, we discussed about the various methods of integration and then we discussed a few methods in qualification also. We discussed about the various methods of qualification basically inspection, modeling, then some instrumental testing as well as few demonstrations also as part of the qualification strategies.

Most of these methods or the qualification strategies are carried out by the system engineers and then you to make sure that whatever they developed, whatever they designed is actually meeting the requirements as specified in the requirements documents. But that alone is not sufficient in order to make sure that the system is satisfying as the requirements, we need to make sure that the system is accepted by the customers and so the last stage in qualification process is the acceptance testing.

So, today we will discuss about the acceptance strategies and acceptance stress methods and how do we plan for the acceptance test and what are the important steps involved in the acceptance testing, what are the things to be tested and how do we plan for its; we will discuss about these aspects in today's lecture.

(Refer Slide Time: 01:42)



Acceptance Testing

- Final test in qualification
- Separated from validation as it is conducted by stakeholders and not system engineers
- If this fails, system has to be changed.
- Thresholds of acceptance: pre-defined
- Focuses on use of system by a small representative group
- Two main questions – What? How?

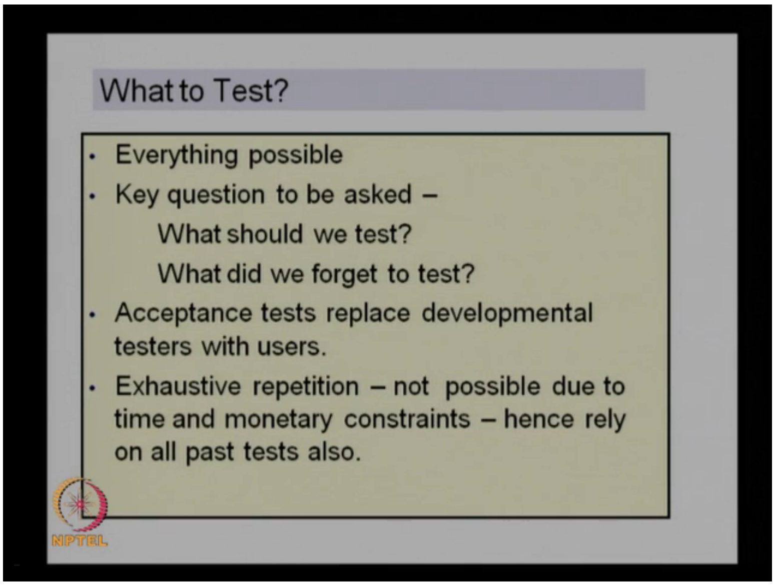
So, acceptance test is basically the final test in qualification. As I mentioned this is separated from the validation as it is conducted by stakeholders and not system engineers and if this fails the system must be changed. So, if the acceptance test fails then definitely then this system has to be changed except for a few exceptions; most of the times we will have to change the system, because this is not acceptable to the customer.

And the threshold of acceptance need to be defined in the beginning itself; so, we cannot have a threshold identified after the system design. So, what is the threshold values for various system design or the system aspects or the system performance need to be defined well before the test and or even before the beginning of the system design, we need to ensure that the thresholds are define.

So, that there is no ambiguity of the thresholds and there is no changes in the threshold after system has been designed or based on the actual system design, we cannot change the threshold; so, it has to be predefined. And acceptance test basically will focus on the use of a system by a small group of people or a representative group because this is not done by the system engineers, we need to have a group of people who will be the users since it is not possible to have a very large group of people testing the system, we need to identify a small group of people who can actually do the testing and then find out whether it actually meets a requirements.

So, various aspects of the system used will be tested by these set of people and then if it meets the threshold values, then it will be accepted. The two main questions asked here is basically what to test, and how to test? So, what are the things to be tested in the acceptance test, and how to do this testing? So, these are the two important aspects in the acceptance testing.

(Refer Slide Time: 03:49)



What to Test?

- Everything possible
- Key question to be asked –
  - What should we test?
  - What did we forget to test?
- Acceptance tests replace developmental testers with users.
- Exhaustive repetition – not possible due to time and monetary constraints – hence rely on all past tests also.

The question of what to test is the first one; basically the answer is everything possible. So, there is no restrictions on this, so whatever the things we can test; we can do the testing that is everything possible can be tested. And the key question should be asked or what should be test? And what did we forget to test? So, this is also very important what should we test is an aspect which we should take care of in the testing, but more importantly what did we forget to test is also important.

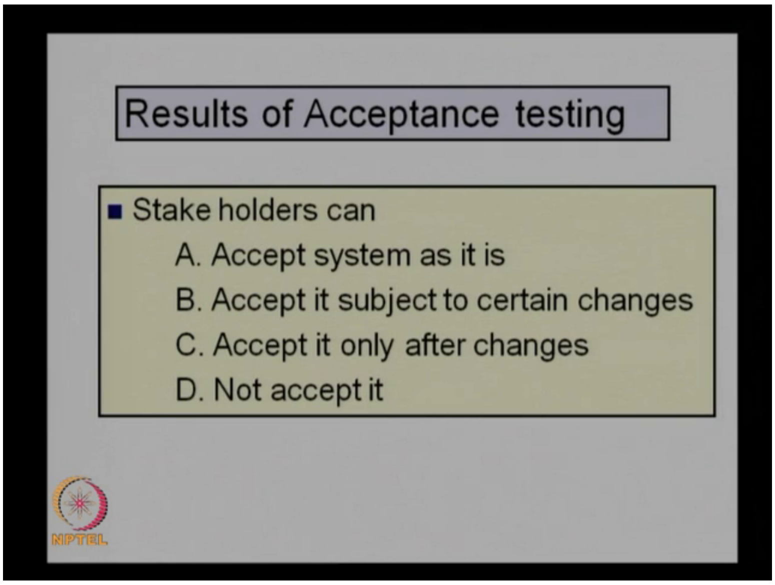
Sometimes we may forget to test some important parameters and that may lead to a failure in the system. There are many case studies which actually showed that has some of the aspects were tested and exactly these parameters became the cause of failure in the system. Therefore, we need to see that it is not only important that what are the things we are testing? But what are the things we forget to test? Also is very important.

So, in this case the acceptance test they replace the development testers with users. So, this I already mentioned; so, the development in testers are different from the users and therefore, and the acceptance test need to be conducted by the users only, this is not done by the developmental testers.

So, that part need to be taken care of and it is not possible to have a exhaustive repetition of the test; mainly because there are lot of test to be conducted and there are many sub system, components and interfaces. So, it is not possible to have an exhaustive test of the system at the acceptance level. So most of the time, we will have to rely on the past test also. So, some of the data from the past test basically in the qualification stage and other sub system integration stages, some of the test data will be used here and that lot of reliance on these data will be there during the acceptance test.

So, depending on the type of the system and depending on the nature of the test requirement, the designers can actually decide; what are the important test to be conducted during acceptance? And what are the data to be used from the previous test? So, that is important here because there are lot of things to be tested. So, this is may not be possible always during the acceptance test.

(Refer Slide Time: 06:07)



Results of Acceptance testing

- Stake holders can
  - A. Accept system as it is
  - B. Accept it subject to certain changes
  - C. Accept it only after changes
  - D. Not accept it

And the results of the acceptance testing; so when we will do an acceptance test basically the idea is to see, whether it can actually meets the requirement of the stakeholders.

So, what kind of results we can obtain from here, after the end of the test stakeholders can actually accept the system as it is. So, that is one option the system can be accepted as it is without any major changes. So, most of the parameters are tested and they actually meet the acceptance threshold, the stakeholders can accept the system as it is.

But then another option is that accept it subject to certain changes. So, there are few changes needed in the system based on the testing acceptance testing the stakeholders can suggest some changes and then accept it based on these changes. So, then in this case the system will be accepted, but they may request for some changes and the stakeholders can start using the system, but changes will be implemented later based on the acceptance test results; that is a second option.

The third option is that accept; it only after changes this, the system cannot be accepted in the present status and deem to be some changes and the system can be accepted only after making the changes. So, in this case compared to the previous one here the system will not be used by the stakeholders, they will wait for the changes to be made and then only it will be accepted and the last option is not too except it.

So, the stakeholders can simply say that it is not meeting the requirements and therefore, we cannot accept the system and it needs to be redesigned. So, there is another option here the system need to be redesigned and then the acceptance test to be repeated and the once that is qualified after the acceptance test, then only it will be accepted by the stakeholders, so these are the four options for stakeholders. So, based on the acceptance testing they will decide either to accept the system as it is because no need of any more changes.

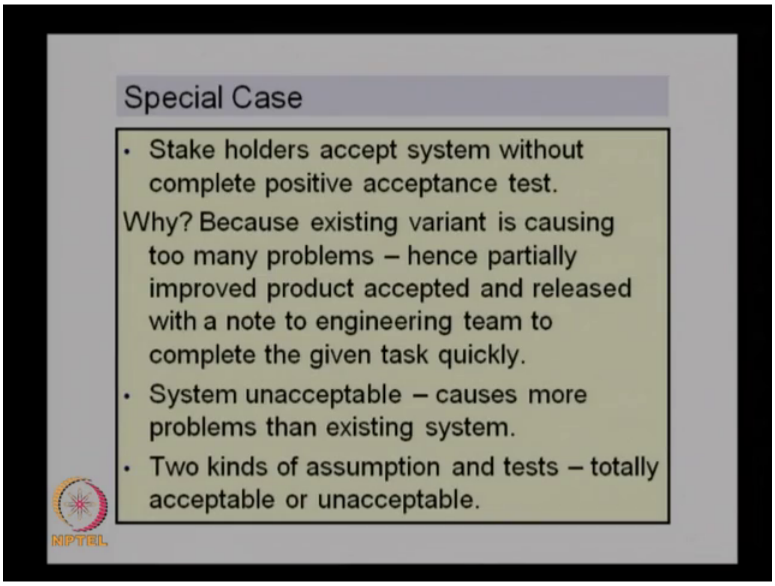
So, in this case the stakeholders can start using the system without any changes, but the second option is that they will ask for some changes or they found out there are some small changes to be made either in the interface or in the functional performance, which are minor and does not really affect the performance of the system.

So, in this case the stakeholders will accept the system as it is with the minor changes. So, the minor changes will be intimated to the designers and designers will make the modifications and add to the existing system. So, the stakeholders will start using the system with even though the minor changes are needed and these minor changes; will come as a next version or a updation of the existing system. The third one is the system meets most of the requirements almost it qualifies in most of the cases, but require some changes which are very important and the system can be accepted only after these changes are made.

So, in this case the stakeholders you know use the system they will ask the designers to make the changes and then give it to them for using. So, that is the third option and the last one is reject the system, so it is not qualifying any of the or most of the requirements are not met by the system. So, in this case they will reject the system and ask the designers to redo the system or redesign the system.

And then it has to go through the integration and qualification stage and the qualification test as well as the acceptance test to be repeated and then it will be accepted. So, these are the various options for the stakeholders after the acceptance test, but there are some cases where there are few acceptance, where the system will be accepted even if there are some problem with the system.

(Refer Slide Time: 10:08)



Special Case

- Stake holders accept system without complete positive acceptance test.

Why? Because existing variant is causing too many problems – hence partially improved product accepted and released with a note to engineering team to complete the given task quickly.

- System unacceptable – causes more problems than existing system.
- Two kinds of assumption and tests – totally acceptable or unacceptable.

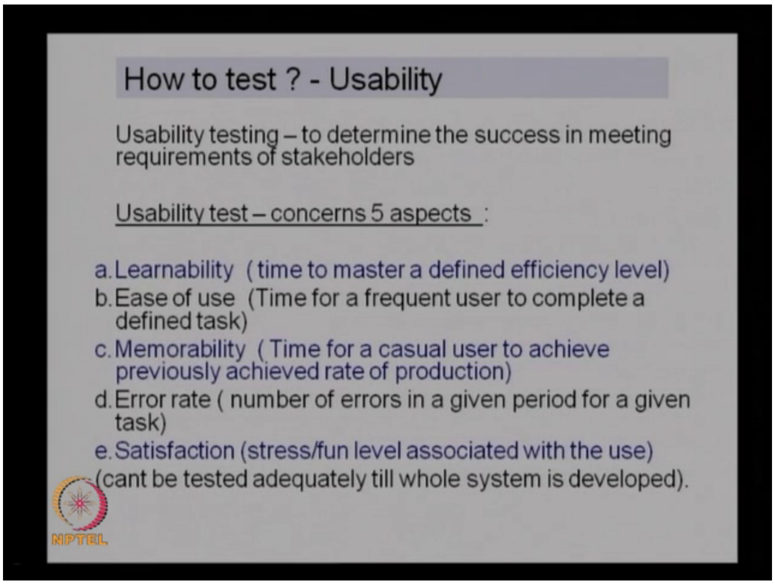
So, here these are the special cases the stakeholders accept the system without complete positive acceptance test. So, here even though the system is not completely satisfying the requirements, the stakeholders will accept the system. This is because the existing variant is causing too many problems, so whatever they are having at present; it is causing too many problems and hence a partially improved product accepted and released with a note to engineering team to complete the given task quickly.

So, here they will accept it because the previous version or whatever is existing at present is causing too much of problems. So, they want to have many improvement upon that one is acceptable to them. So, they will start using it, but they will give a note to the design team to ask him to change the design or asking to improve it. And system is unacceptable in some cases though it is meeting the most of the requirement; it is not acceptable because it causes more problems than existing system.

So in that situation they may not accept it, though it actually meets the many of the requirement. So, if it causing more problem than the present system then it will not be accepted and in most of these test, they will make two assumptions and then go ahead with the testing. Basically one is that; they assume that it is totally acceptable or unacceptable. So, if this is assume it is totally acceptable then most of the test will be designed in such a way that to prove that it is not acceptable.

So, you assume that is unacceptable and then conduct test to prove that, it is not acceptable and if this test fails; then it is acceptable. Similarly, you assume that it is unacceptable and then carry out the test to prove that it is acceptable and if the test succeeds then the system will be accepted. So, this way we can have two kinds of assumptions and test to carry out the acceptance test.

(Refer Slide Time: 12:04)



How to test ? - Usability

Usability testing – to determine the success in meeting requirements of stakeholders

Usability test – concerns 5 aspects :

a. Learnability (time to master a defined efficiency level)
b. Ease of use (Time for a frequent user to complete a defined task)
c. Memorability (Time for a casual user to achieve previously achieved rate of production)
d. Error rate (number of errors in a given period for a given task)
e. Satisfaction (stress/fun level associated with the use) (cant be tested adequately till whole system is developed).

The next one is how to test? So, that there are various ways of testing and various things to be tested and there are different options for the stakeholders after the test. But how to do the testing is again an important aspect because there are various aspects to be tested and how do we carry out these test is important. So, here we look at one aspect as the usability of the system and then see how to do the testing of for usability.

So, usability testing is to determine the success in meeting requirements of stakeholders. So basically the stakeholder requirements; how much it is satisfied, how the usability of the system by the stakeholders is satisfied by the system is tested. There are five aspects in the usability test; basically you look at the learnability of the system that is time to master a defined efficiency level.

So, we look at the system and then see how first can be learned by the or mastered by a user. We will have a predefined efficiency level and we will see how much time it takes for a user to reach that efficiency level; that is the learnability test. The next one is the ease of use; now how ease to use the system, so here time for a frequent user to complete a define task.

So, you find out the time required by a frequent user to complete a defined task. So, there are users with various capabilities; some of them are very frequent user, some of them rarely use the system, some of the experts in the use of the system. So, we take a medium level user, a frequent user to complete a define task and find out how much time he takes to complete a particular task.

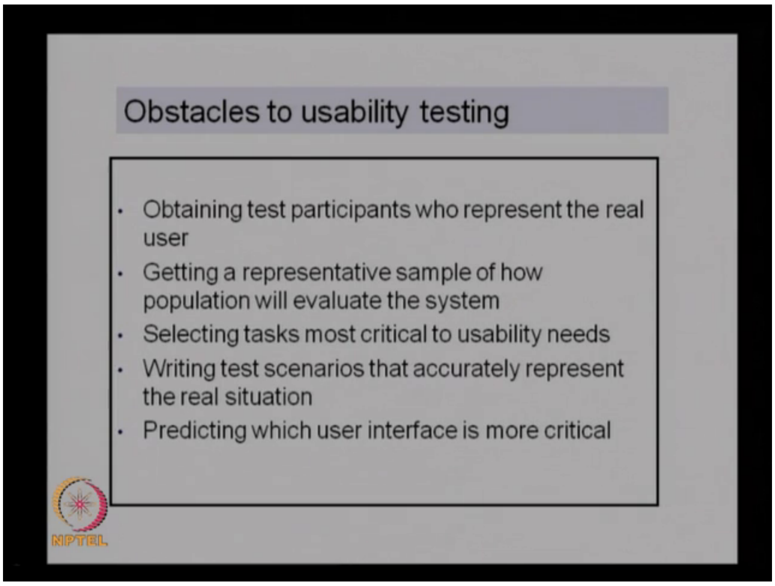
So, that actually gives us the use of the system and the third one is memorability of the system. So, time for a casual user to achieve previously achieved rate of production. So, here it is a casual user is not a every frequent user, so he will be asked to use the system and to see how much time it takes to achieve a previously achieved rate of production.

So, he may be having some level of production using a previous version of a system or a similar system. So, this user will be asked to use the system and to see how fast he actually reaches the normal rate of production and that actually gives us the value of the memorability of the system. And the fourth one is error rate, the number of arrays in a given period for a given task. So, depending on the number of errors we can find out what is the error rate of the system? And then for a given period, for a given task we will try to find out how many errors are happening in the system and that actually gives an error rate of the system.

Then the satisfaction level, the satisfaction level is the again it is a little bit subject you, but then it is the stress or fun level associated with the user. So, basically a person who is using the system; how happy he is or how much troubled he is or how much frustrated he is; on using the system. So, that is a measure of the satisfaction of the system; so, the usability of the system can be tested using this five measures.

And the last one; that satisfaction level can be tested only when the complete system is available. So, when we have the complete system; we can do the satisfaction test, but the other, the test can be completed even when the complete system is not available also. So, these are the five levels of usability testing and we can actually take these parameters and measure their values and see how usable the system is and what are the obstacles basically there in the usability testing.

(Refer Slide Time: 15:34)



Obstacles to usability testing

- Obtaining test participants who represent the real user
- Getting a representative sample of how population will evaluate the system
- Selecting tasks most critical to usability needs
- Writing test scenarios that accurately represent the real situation
- Predicting which user interface is more critical

Though there are factor emitted, there are some problem in carrying out the usability testing. So, these are basically the first one is of course, getting the test participants; who represent the real user? So, we need to have a set of users to do the testing, but then it may not be always easy to find the users or the test users the function, number of test users who can actually participate in the testing of the system. So, that actually gives a problem and we need to ensure that who are we selecting for the testing of the system; they represent the real users or there is a chance for them to use or there is a potential for these people using the system.

So, then only there will be some validity for the test. So, getting these people is bit difficult because finding them for the testing or getting them for that testing to come to the test facility and do the test may not be always possible. So, that is one of the problems in doing the usability test and then getting a representative sample of how population will evaluate the system; that is we need to have a group of people, who actually can evaluate the system using one person or a few numbers may not be sufficient because we need to get a representative sample of how population will evaluate the system.

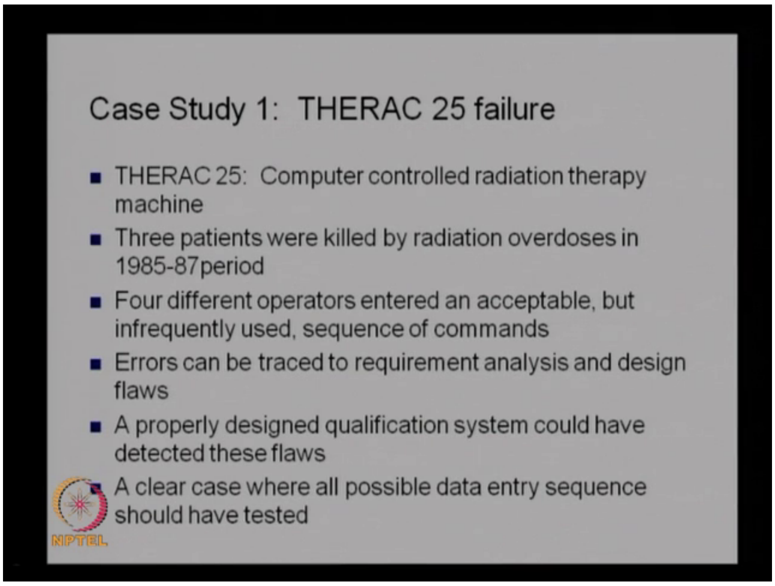
There may be various sections of people using the system; there may be a system, may be used by adults or may be children or physically challenged people or those who are elderly. So, we need to get a representative sample of all these people to evaluate the system that also might be a difficult task. Then selecting tasks most critical to usability needs, so as I mentioned we need to see the task; which are needed to be tested by the users but then this selecting the task most critical to usability needs also is important.

So, the designers need to analyze all the task and then see which are the most critical task and accordingly they need to do the testing. And then writing the test scenarios the accurately represent the real situation. So, the scenarios need to be properly defined and written down so that all the test can be done based on these scenario. So, writing this scenarios also; a big challenging and predicting which user interface is more critical. So this again difficult to predict initially because depending on the use and that depending on the people who are using, so some of the interface maybe critical then just many not be critical.

So, identifying this critical interface also a challenge, so these are the main obstacles in doing the testing. So, we need to ensure that whenever we do a usability, we need testing; you need to make sure that we have a representative, a sample of the population who use the system and we have the scenarios identified properly and we have the critical user interfaces identified and we write down the scenarios and to test and what are the important task to be tested.

So, all these need to be analyzed and recorded properly to make sure that we do a proper testing and that those testings are really important. And these test values really represent the acceptance of the system by the user. So, though these challenges in the usability testing need to be identified early stages and measures to be taken to overcome these difficulties; so that is about the obstacles in usability testing.

(Refer Slide Time: 19:12)



Case Study 1: THERAC 25 failure

- THERAC 25: Computer controlled radiation therapy machine
- Three patients were killed by radiation overdoses in 1985-87period
- Four different operators entered an acceptable, but infrequently used, sequence of commands
- Errors can be traced to requirement analysis and design flaws
- A properly designed qualification system could have detected these flaws
- A clear case where all possible data entry sequence should have tested

So, as I mentioned these are the various stages in the acceptance testing. So, we have to identify the test scenarios, we need to identify the test scenarios as well as we need to identify the task to be tested and so many of the test decayed out that usability testing. How do we do usability testing? So, here I will take two case studies to show that how important is the testing and acceptance test in the system qualification. We discussed about the case of THERAC 25 failure, but this was a medical, a device; a computer controlled radiation therapy machine and developed for giving controlled radiation to the patients and then in the period of 1985 to 87; 3 patients were killed by radiation overdoses and though machine was supposed to protect these patients, they were killed because of the overdoses in radiation.

So, the reason for this was the four different operators entered an acceptable, but in frequently used sequence of commands. So, the reason for the overdose was basically the use of the system by various operators. So, four different operators entered an acceptable, but in frequently used sequence of commands and that actually led to the overdose and these errors can be traced to requirement analysis and design flaws.

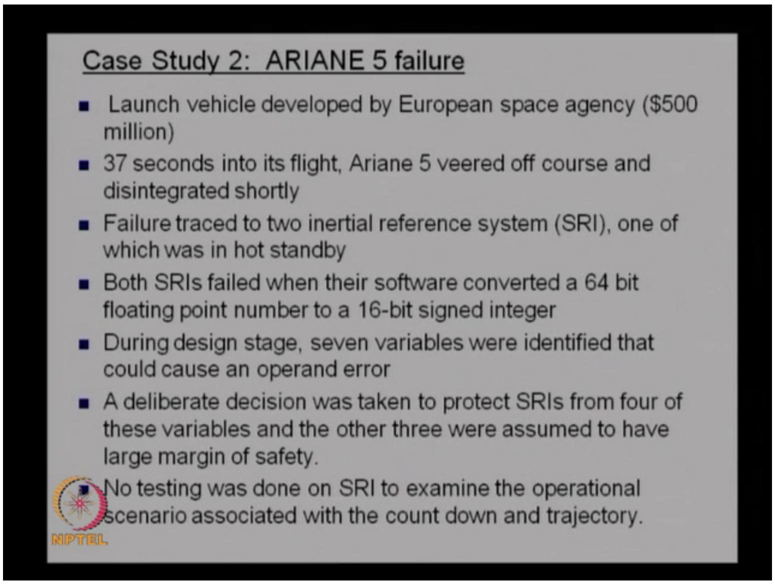
So, when we start the designing of the system it was; as I mentioned earlier we need to identify the requirement properly and then record them. So, there was an error in identifying the requirement, the proper requirement were not identified. What is the condition under which these parameters to be entered? And what is the sequence in which need to be entered? Those things were not properly identified and the properly designed qualification system could have detected these flaws.

So, though there were errors in the beginning in the analysis stage or the requirement analysis stage, if we had a properly designed qualification system then these flows could have been awarded or it can be could have been detected in the initial stage or before implementing the machine or before deployment of the machine, we could have identified these flaws. So, the proper testings were not carried out; where we can identify these requirement or these scenarios.

So, the operating scenarios like the; in frequently used scenario could have been simulated and formed the testing could have been carried out to find out what actually, what will be the output and then the errors could have been identified and then could have been rectified. So, that was the problem really what actually caused the tragic death of many patients because of overdose.

So, this was a clear case where all possible data entry sequence should have tested. So, this actually shows the importance of acceptance testing. So, if all the possible data entry sequence should have been tested to make sure that there cannot be error in data entry sequence and which may cause the system to view some errors. So, this actually shows the importance of acceptance testing and identifying all the scenarios of testing and this is one case study that is the another one again from the industrial accidents or the failures.

(Refer Slide Time: 22:39)



Case Study 2: ARIANE 5 failure

- Launch vehicle developed by European space agency ($500 million)
- 37 seconds into its flight, Ariane 5 veered off course and disintegrated shortly
- Failure traced to two inertial reference system (SRI), one of which was in hot standby
- Both SRIs failed when their software converted a 64 bit floating point number to a 16-bit signed integer
- During design stage, seven variables were identified that could cause an operand error
- A deliberate decision was taken to protect SRIs from four of these variables and the other three were assumed to have large margin of safety.
- No testing was done on SRI to examine the operational scenario associated with the count down and trajectory.

The ARIANE 5 failure, we discussed about this in one of the lectures. So, ARIANE 5 was the launch vehicle developed by the European space agency, it causing around 500 million US dollar and 37 seconds into its flight ARIANE 5 veered off course and disintegrated shortly.

Again the failure was traced to two inertial reference systems, one of which was in hot standby, we lend about the standby system which basically it is a fall tolerant and system. So, there were two initial reference systems; one of which was in hot standby, so even if one fails, the other one can be used to get the output from the system and then can find out the location of the system the vehicle.

Both SRI's failed when their software converted 64 bit floating point number to a 16 bit signed integer. So, this was the basic cause for the failure there was a conversion of a 64 bit floating point number to a 16 bit signed integer which was not planned in the system or they never expected this kind of a conversation. And since it was converted into this and the system could not accept, the interface could not accept the data and therefore, there was a failure.

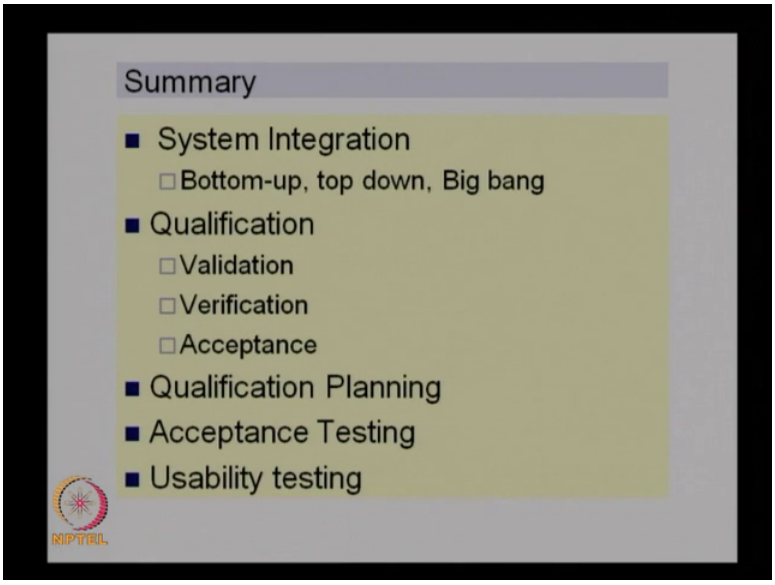
Now, during the design stage seven variables were identified that could cause an operand error. So, in the initial stages of the design; the design engineers have identified seven variables which can actually cause this kind of a problem and they identified four of them. So, a deliberate decision was taken to protect SRIs from four these variables; so, out of the seven, they identified four variables and they made necessary measures to protect the system or the SRI's from this problem. And four of them have been protected and the other three were assumed to have a large margin of safety.

So, again they to get decision here saying that there is a possibility of these three getting into this there are resulting into a operand error is very remote and then they are decided not to protect them against this three variables and no testing was done on SRI to examine the operational scenario associated with the countdown and trajectory. So, the SRI's is the basically used during the initial stage of the flight. So, they did not do any acceptance test to see whether this kind of a scenario will be generated or not.

So, since they have identified three parameters or three variables which can actually cause an opponent error, they could have tested these conditions to ensure that this will not cause a problem. But unfortunately exactly one of these variables resulted into a an operand error and that actually caused the problem in the whole launch, vehicle was destroyed by that simple issue and later on when they changed that one and then they identified the reason for the failure, they could rectify it very easily and then solved the problem.

But if there was a proper testing which actually examined the particular scenario, where these three variables may result into an operand error; then this failure could have been eliminated. So, this again shows the importance of a acceptance testing in the system and identifying all the possible scenarios of developing an error. And then making sure that the acceptance test actually ensured that such errors will not happened and it actually qualifies through this acceptance test. So, once it is qualified then the system is ready for deployment.

(Refer Slide Time: 26:20)



Summary

- System Integration
  - □ Bottom-up, top down, Big bang
- Qualification
  - □ Validation
  - □ Verification
  - □ Acceptance
- Qualification Planning
- Acceptance Testing
- Usability testing

So, in the last three lectures we discussed about the system integration and qualification and here we mentioned that, we have various stages in integrating the system. So, we start with the subsystems and then components and then subsystem. Then interfaces, we start integrating them and there are different methods of integrating, the system that is basically, the bottom up approach, top down approach, then big bang approach; so we can use any of these methods to integrate the system.

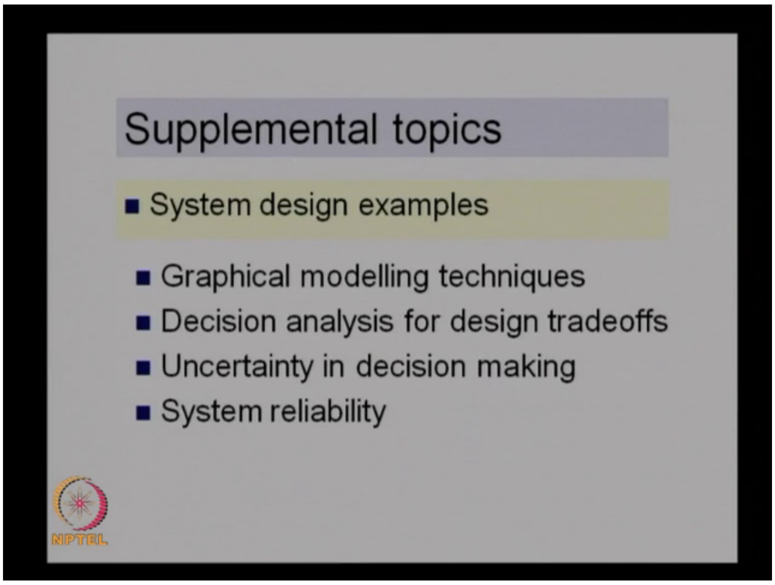
Similarly, we discussed about the qualification strategies basically the validation, verification and acceptance. So validation and verification are basically carried out by the system designers and the acceptance test is basically carried out by the users with the help from the designers. So, designers will identify the task to be tested and the scenarios to be tested and then the users will be asked to test them and then based on the; for particular threshold value, the system will be accepted.

So, the users have different options; after testing they can actually accept the system or they can accept it with some changes or they can reject the system. In some cases and some special cases, the users will be forced to accept the system even if there are some errors because the existing system is giving lot of troubles. So, they want avoid these troubles and then you continue to; they want to new system, just to escape from the errors of the previous system. These are the options for the users and then we discussed about the acceptance testing procedures at the usability testing procedures. And then we saw few case studies, where failures in acceptance test caused the system failure.

So, these were the topics we discussed in the last three lectures and as I mentioned in the previous lectures, so this is the last step in the design process that the six functions of the design process. So, if we complete these then actually for each life cycle, we have to complete these six functions of the design process. In the next few lectures, we will discuss about few supplemental topics which are actually useful in the system design. So, before going for the supplemental topics; I will take few case studies or few system design examples to show you; how do we actually use the principles, whatever we learnt in a real scenario or real system design scenario and how to go in a systematic way of designing the system.

So, I will take three case studies and explain the procedure and we will not be going into the details of each and every stage, but I will be giving an overall view of how the principles can be used in the system design.

(Refer Slide Time: 29:04)



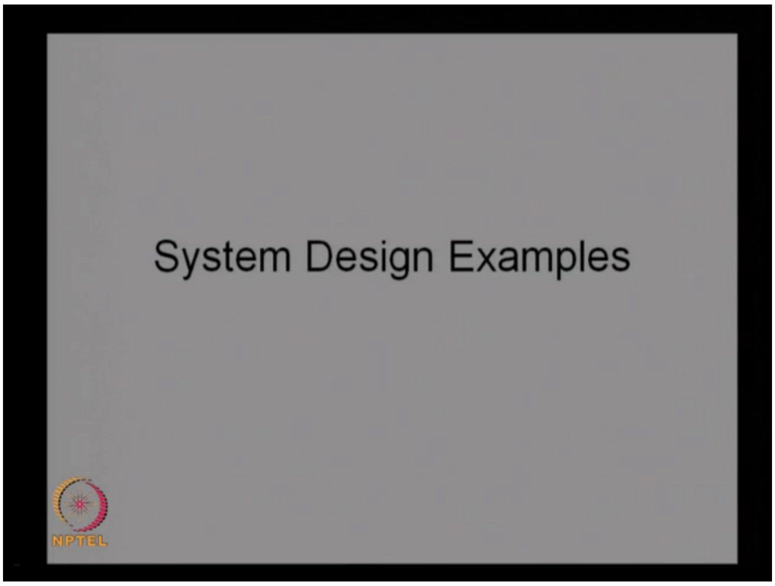
Supplemental topics

- System design examples
- Graphical modelling techniques
- Decision analysis for design tradeoffs
- Uncertainty in decision making
- System reliability

And after that, we will be going to the other supplemental topics like graphical modelling and modelling methods, decision analysis; decision making and decision analysis tools. Then we will look the reliability of the system; how do they design the system for reliability. Similarly, how to avoid the faults in the system? How do we incorporate the fault analysis into the system design? And similarly we look at some of the statistical tools which can be used in the system design like design of experiments and other methods.

To start with, we will go for the system design examples; now I will take few examples here and then explain the procedure of a system design using the standard system engineering principles.

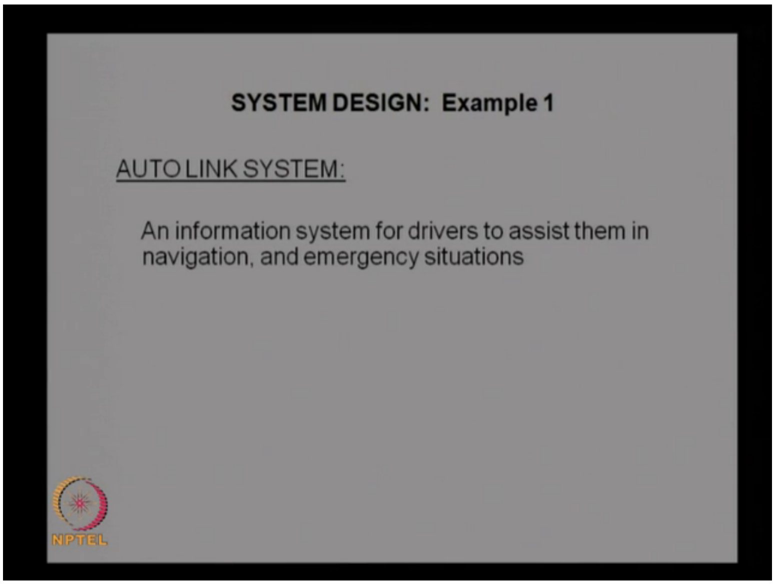
(Refer Slide Time: 29:53)



System Design Examples

So, this is the first example for the system design.

(Refer Slide Time: 29:57)



SYSTEM DESIGN: Example 1

AUTO LINK SYSTEM:

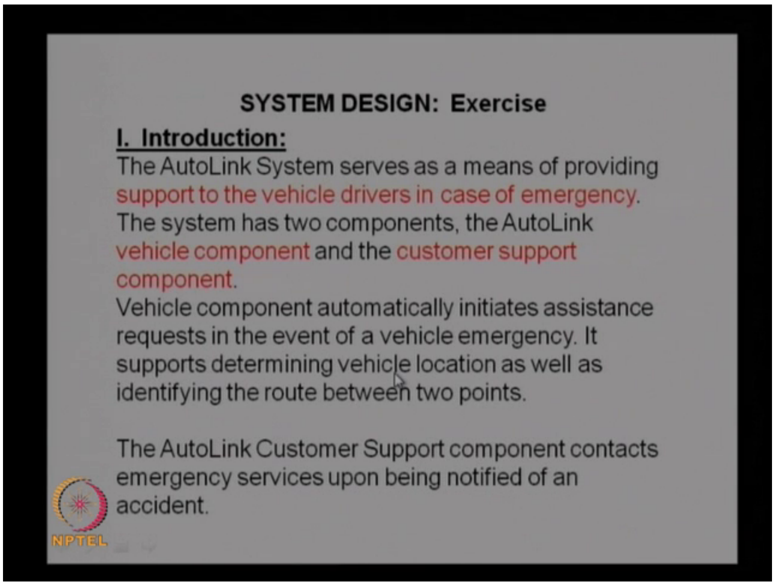
An information system for drivers to assist them in navigation, and emergency situations

We will take the case of an auto link system, I mentioned about this in one of the earlier lectures while explaining one the functional decomposition. I will explain about the auto link system, so we will look into the detail of this system and then how do we actually start with the system design, with the procedures or the methods we already discussed.

The auto link system is an information system for drivers to assist them in navigation and emergency situations. So, we are actually trying to design a system which can actually be used by the drivers or passengers in a car or any other vehicle in an emergency situation or to get some other information about the navigation aspects.

So, if somebody is driving on a highway or traveling on a long distance and they want to get some information about the navigation aspects, the location; their location and to find out the route to another location or there is an emergency situation in the car, there is an accident or there is a theft or there is some other issues; then how do I contact a centre and get the help from the system; so that kind of information system is the auto link system, so basically linking the automobiles to a central location or to the emergency service providers.

(Refer Slide Time: 31:19)



SYSTEM DESIGN: Exercise

I. Introduction:

The AutoLink System serves as a means of providing support to the vehicle drivers in case of emergency. The system has two components, the AutoLink vehicle component and the customer support component.

Vehicle component automatically initiates assistance requests in the event of a vehicle emergency. It supports determining vehicle location as well as identifying the route between two points.

The AutoLink Customer Support component contacts emergency services upon being notified of an accident.

So, here the auto link systems are as a means of providing support to the vehicle drivers in case of emergency. The system has two components, the auto link vehicle component and the customer support component.

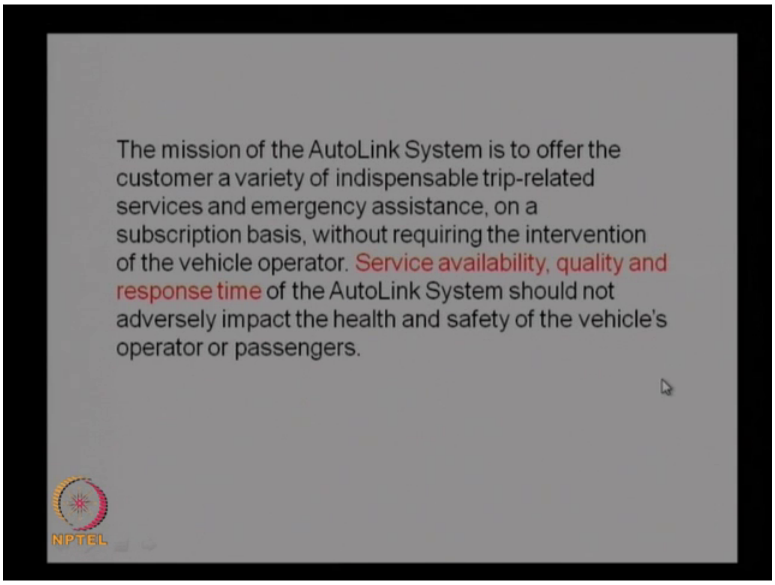
So, there are two important components in the system; one is the vehicle component, the other one is the customer support component; the vehicle component will be having its on hardware and software. And again customer support will be having its on hardware and software and they will be linked together to provide the service. So, main design focus will be on these two component vehicle component and the customer support component, the vehicle component automatically initiates the assistance request in the event of a vehicle emergency; its supports determining vehicle location as well as identifying the route between two points. So, the role of the vehicle component is basically to initiate the emergency assistance request.

So, either it can be initiated by the driver or it can be initiated automatically and then it supports determining the vehicle location as well as identifying the route between two points. So, it actually assist in the other part also the navigation part also; it will support the determining the vehicle location as well as identifying the route between two points; the customer supports component contacts emergency services upon being notified of an accident.

So, the main role of the auto link customer support is basically to contact the emergency services upon being notified of an accident. So, they can contact the ambulance service or the police service depending on the situation and that part will be done by the customer supports. And of course, it can actually help the first other aspect also basically on the vehicle location as well as route between two points.

So, in case of a failure in the other systems then the customer support will be able to provide necessary assistance also. So, the primary task of the customer support is to contact the emergency services.

(Refer Slide Time: 33:17)



The mission of the AutoLink System is to offer the customer a variety of indispensable trip-related services and emergency assistance, on a subscription basis, without requiring the intervention of the vehicle operator. Service availability, quality and response time of the AutoLink System should not adversely impact the health and safety of the vehicle's operator or passengers.

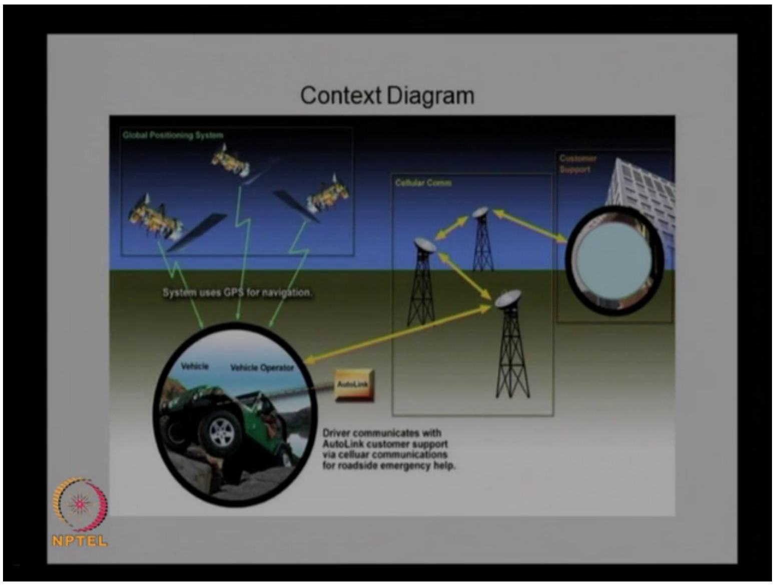
So, how do we actually start the design? Also you have to look at the actual design of mission of the system. So, the mission of the link is to offer the customer a variety of indispensable trip related services and emergency assistance on a subscription basis without requiring the intervention of the vehicle operator service availability, quality and response time of the auto link system should not adversely of impact the health and safety of the vehicles operator or passengers.

So, when we start designing we need to clearly state the mission of the system and what are the limitations and the prospects of the system. So, here it we are clearly stating that it actually provides a variety of indispensable trip related services, an emergency assistance on a subscription basis. So, it is not a free service; so somebody have to ask to subscribe to the particular service without requiring the intervention of the vehicle operator.

So, the vehicle operator need not keep on intervening to the system. So, automatically it will provide the services and at the same time, the service availability quality and the response time of the system in no way should affect the health and safety of the vehicles operator or passengers. So, that is the limitation here; not the limitation basically we are putting a constraint over here. So, what are the service provided and its quality and response time should not in any way effect the health and safety of the vehicles operator or passengers.

So, that is the condition under which the services to be provided.

(Refer Slide Time: 34:40)



Context Diagram

Driver communicates with AutoLink customer support via celluar communications for roadside emergency help.

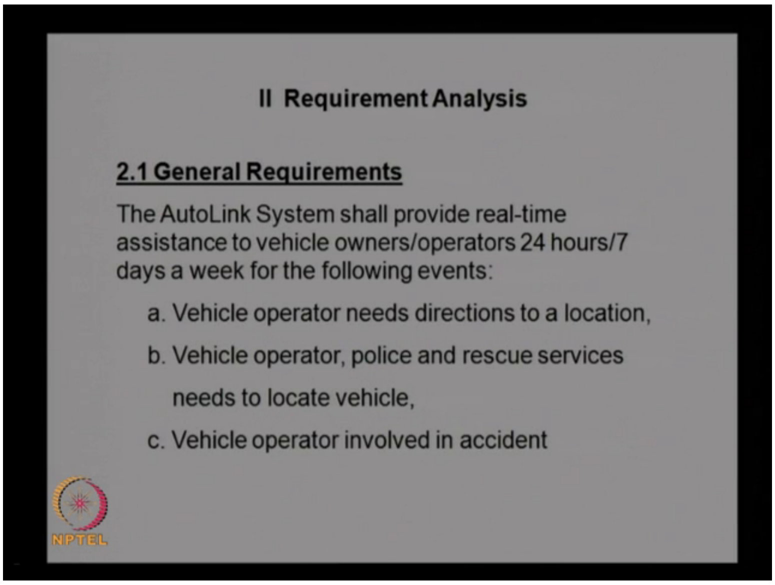
So, we need to look at the context diagram; so, what will be the context diagram for this particular system? So, how do we actually start with the system design? So, we will develop a initial context diagram in under which the system will be operated as we can see here, this will be the operator; the customer support component will be located in a different location may be in a nearby city or a town and there will operators, will be operating or will be responding to the request from the vehicle and this is the vehicle and these vehicle component will be inside the vehicle and there will be a navigation system accessories as well as software and hardware and that will be communicating with the GPS system satellites.

So, this will be used to get the location of the vehicle and based on this information; it will be able to localize the vehicle and identify its location and then there will be a communication with customer support. So, it can be a mobile communication; so a wireless communication from the vehicle to the customer support. So, that is the contacts in which the system will be operating.

So, what you need is the GPS data, the global positioning system data then the vehicle components which can actually identify these, get these data and then localize and then the communication from the vehicle to the operator. So, these are the basic requirement for the system and then the auto link system will be there in the vehicle and it will accept the data and transmit the data, whenever it is needed to the customer support.

What we need to design is the auto link system, so we do not need to really worry about the GPS system or the cellular system. We assume that these are already existing, what we need to design is the auto link system for the vehicle as well as the customer support system, what are the parameters or what are the requirement at this customer support as well as the auto link we look at this requirements and then design the whole system.

(Refer Slide Time: 36:39)



II Requirement Analysis

2.1 General Requirements

The AutoLink System shall provide real-time assistance to vehicle owners/operators 24 hours/7 days a week for the following events:

a. Vehicle operator needs directions to a location,

b. Vehicle operator, police and rescue services needs to locate vehicle,

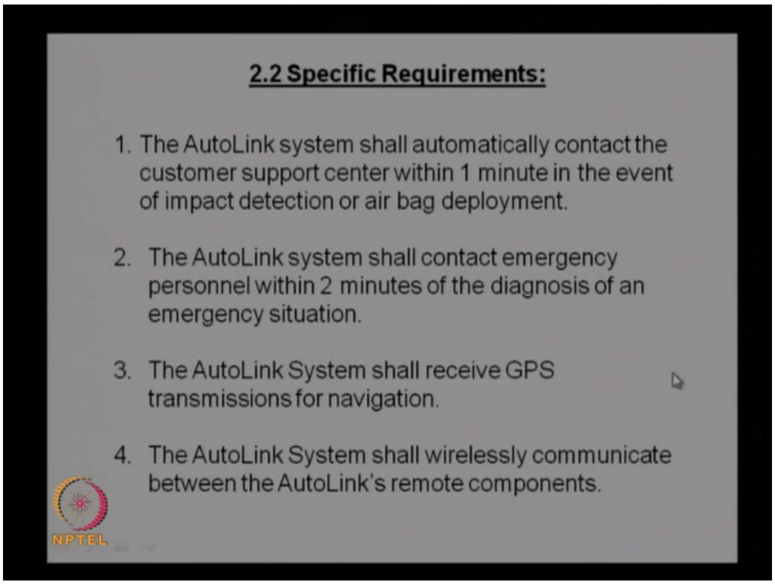
c. Vehicle operator involved in accident

So for that we need to have the requirement analysis; so we will look at the customer, the requirements of the system. So, the general requirements will be analyzed of course, there can be many requirements; I have not written down all the requirements here, some sampled requirements are given here we need to look at the whole system and then develop the complete requirements.

The auto link system shall provide real time assistance to vehicle owners operators; 24 hour 7 days a week for the following events. So, these events are actually identified by the designer; so it should provide 24 hours 7 days a week service. Vehicle operator needs directions to a location; so that is one scenario. The vehicle operator police and rescue services needs to locate the vehicle, so in case of an emergency the vehicle operator police and rescue services should be able to locate the vehicle.

So, identify the location of the vehicle and the vehicle operator involved in an accident. So whenever there is an accident in the vehicle, operator vehicle then the vehicle that scenario at the service should be provided for that scenario also. So, assistance should be provided and for these scenario, for these events.

(Refer Slide Time: 37:49)



2.2 Specific Requirements:

1. The AutoLink system shall automatically contact the customer support center within 1 minute in the event of impact detection or air bag deployment.
2. The AutoLink system shall contact emergency personnel within 2 minutes of the diagnosis of an emergency situation.
3. The AutoLink System shall receive GPS transmissions for navigation.
4. The AutoLink System shall wirelessly communicate between the AutoLink's remote components.

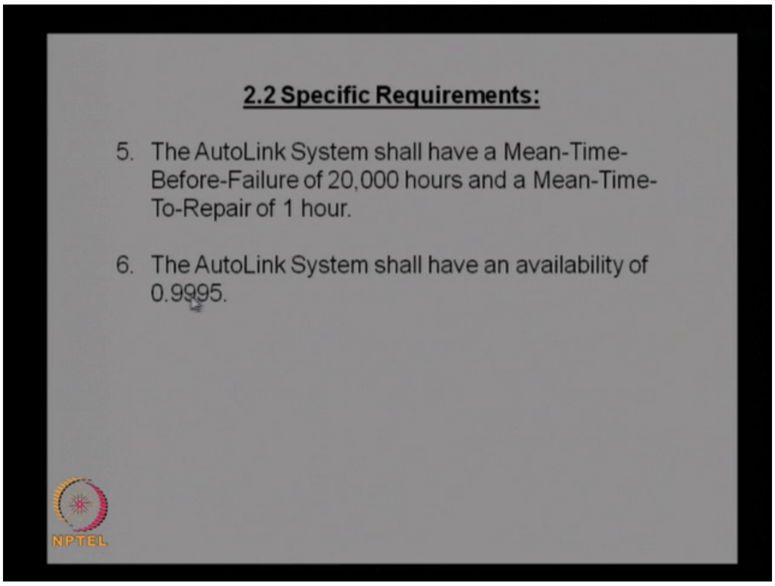
Then the specific requirements; so those were the general requirement. The specific requirements are related to the specific system components, the first requirement is the auto link system shall automatically contact the customer support center within 1 minute in the event of impact detection or air bag deployment.

So, whenever there is an accident the system should immediately contact within 1 minutes, it should contact the customer support centre. So, this contact can be initiated when there is an impact detection or an air bag deployment. So, we can see that this is the first requirement again you can have additional requirements like that, there should be a system to identify the impact and then there should be a time delay. So, the system should identify an impact within a particular time.

Similarly, the system should identify the airbag deployment and communicate this to the customer support centre. So, again additional requirements can be identified from here the other one is the auto link system shall contact emergency person within 2 minutes of the diagnosis of an emergency situation. So, within 2 minutes of a emergency situation; it will be able to contact the emergency personnel. So, this is actually this is requirement from the vehicle to the customer support and this is from the customer support to the emergency personnel. So, within 2 minutes of the diagnosis of an emergency situation, the information should be passed to the emergency personal and the auto link system shall receive GPS transmissions for navigation.

So, it should be able to receive the GPS data for navigation purpose. So, there should be a GPS receiver in the system, the auto link system shall wirelessly communicate between the auto links remote components. So, this is another requirement; there should be a wireless communication between the remote compound; basically the vehicle component as well as the customer support component, there should be a wireless communication. So, these are the specific requirements you can identify.

(Refer Slide Time: 39:47)



2.2 Specific Requirements:

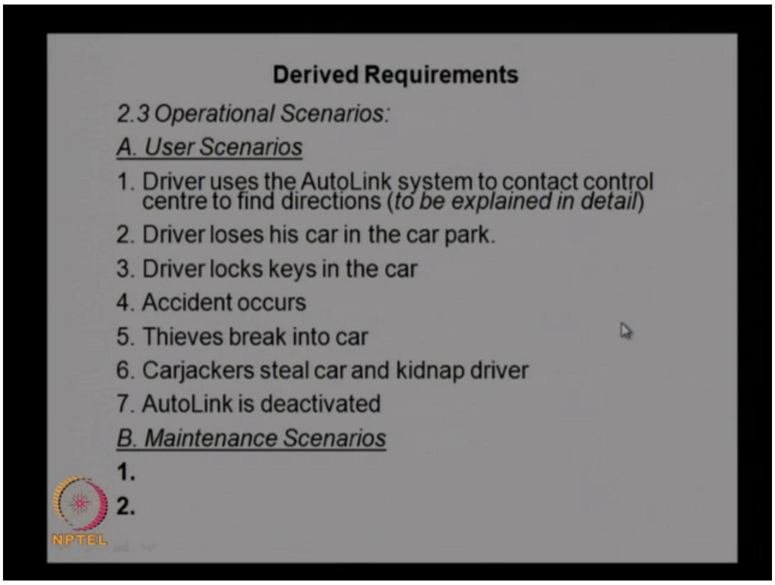
5. The AutoLink System shall have a Mean-Time-Before-Failure of 20,000 hours and a Mean-Time-To-Repair of 1 hour.

6. The AutoLink System shall have an availability of 0.9995.

Similarly, we can have for many requirements like this. So, auto link system shall have a mean time before failure of 20000 hours and a mean time to repair of 1 hour and then the auto link system shall have an availability of 0.9995.

So, like this you can actually go for many requirements; I have not specified all the requirements.

(Refer Slide Time: 40:07)



Derived Requirements

2.3 Operational Scenarios:

A. User Scenarios

1. Driver uses the AutoLink system to contact control centre to find directions (*to be explained in detail*)
2. Driver loses his car in the car park.
3. Driver locks keys in the car
4. Accident occurs
5. Thieves break into car
6. Carjackers steal car and kidnap driver
7. AutoLink is deactivated

B. Maintenance Scenarios

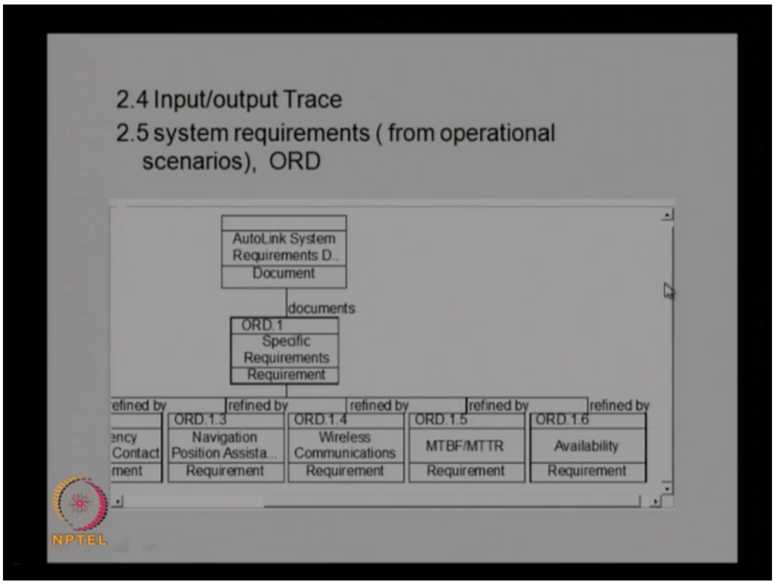
1.
2.

Then based on these requirements, the previous requirements you can go for the derived requirements. So, we can identify many operation scenarios and identify the derived requirements from the scenarios also. So, we can have many user scenarios like driver uses the auto link system to contact control centre to find directions. So, that is one scenario we can actually explained it in detail; what is the input? What is the output?

Then driver loses his car in the car park, so he wants to identify the location of the car, so that can be another application; the driver locks keys in the car. Another application you can think of when accident occurs, there is an accident or thieves break into car; how can we use this particular system for that scenario? Or carjackers steal car and kidnap driver. So, in this situation how the system should have the passenger or the driver; the auto link is deactivated.

So, there may be a scenario when the auto link is getting deactivated. So how the system should react to such scenarios? So, we can identify all these scenarios can get the details of the scenario, like the scenario tracing what we discussed earlier; we can try the scenarios and we can get the inputs and outputs and based on that we can get many of the derived requirements. Similarly, there can be many other scenarios like maintenance scenarios, then there can be training scenarios of course, like that we can have a many scenarios and identify all the requirements.

(Refer Slide Time: 41:30)



2.4 Input/output Trace

2.5 system requirements (from operational scenarios), ORD

AutoLink System Requirements D.. Document

documents

ORD.1 Specific Requirements Requirement

efined by | refined by | refined by | refined by | refined by

ency Contact ment

ORD.1.3 Navigation Position Assista... Requirement

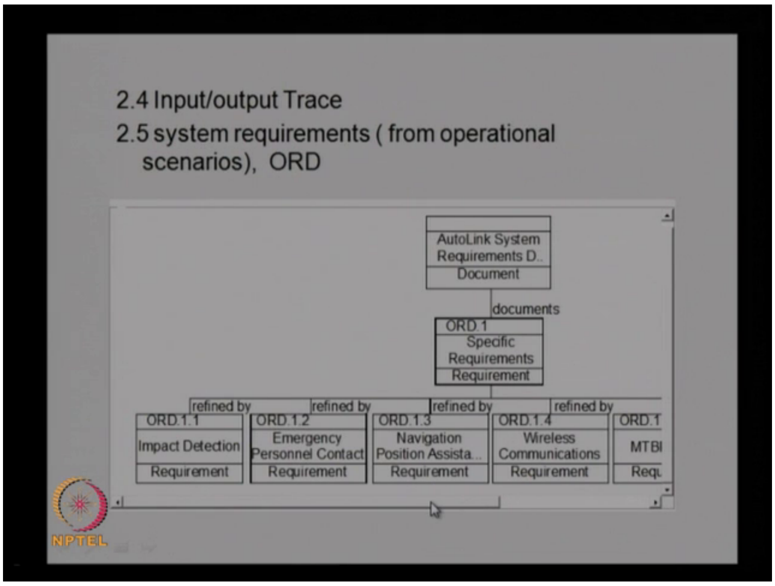
ORD.1.4 Wireless Communications Requirement

ORD.1.5 MTBF/MTTR Requirement

ORD.1.6 Availability Requirement

This is actually shows the requirement documents, so as I mentioned earlier we need to prepare and originating requirement document based on the other requirements analyzed; identified.

(Refer Slide Time: 41:40)



2.4 Input/output Trace

2.5 system requirements (from operational scenarios), ORD

AutoLink System Requirements D.. Document

documents

ORD.1 Specific Requirements Requirement

refined by | refined by | refined by | refined by

ORD.1.1 Impact Detection Requirement

ORD.1.2 Emergency Personnel Contact Requirement

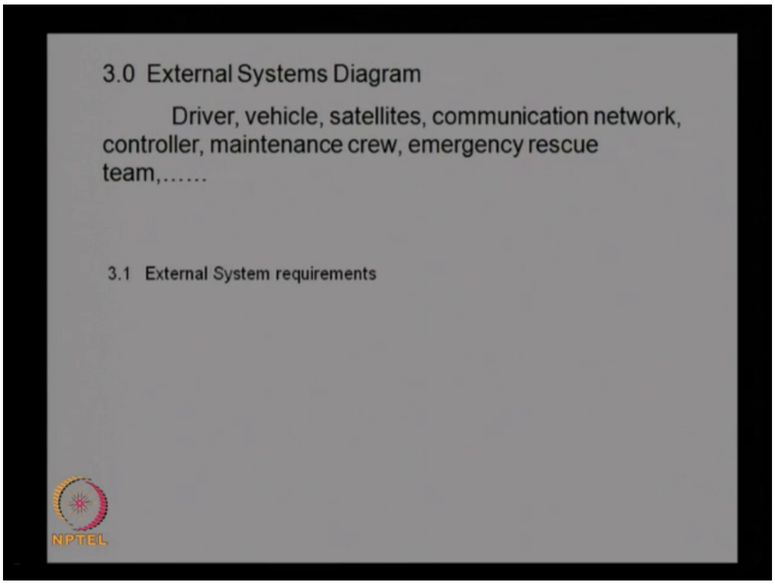
ORD.1.3 Navigation Position Assista... Requirement

ORD.1.4 Wireless Communications Requirement

ORD.1 MTBI Requ

So, we have the main requirements like the impact detection requirement, emergency personnel contact requirement, navigation position assistance, wireless communication requirements then maintain between failure and repair requirement, availability requirement like this you can have many other requirements also depending on all the scenarios identified. We will be having many additional requirements over here, so we will write down those requirements related to impact detection, sub requirements written down over here and all this will be documented properly which becomes the basic document for the design or the originating requirements document.

(Refer Slide Time: 42:26)



3.0 External Systems Diagram

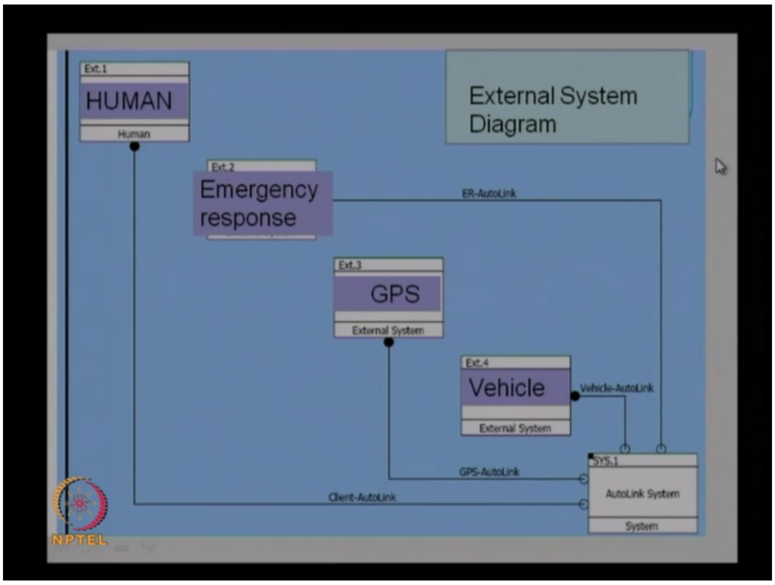
Driver, vehicle, satellites, communication network, controller, maintenance crew, emergency rescue team,……

3.1 External System requirements

Once we have the originating requirements document, we can go for the external system diagram, we can develop the external system diagram based on the external system. There can be many external systems like driver, vehicle, satellites, communication network, controller, maintenance crew emergency, rescue team.

So, these are all the external systems interacting with the auto link system. So, we will identify that all the external systems and the interaction between these external systems.

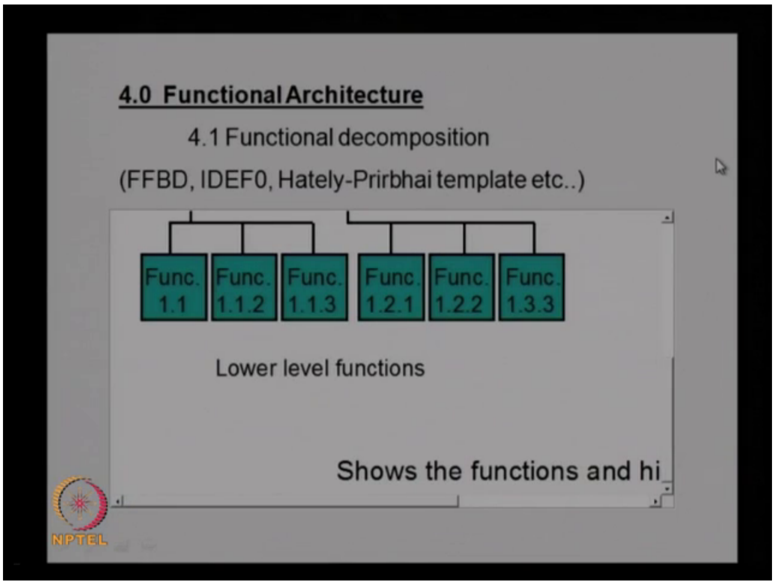
(Refer Slide Time: 42:50)



External System Diagram

So, we can see here these are the main external systems of course, you can identify more like human operator, then you the emergency response team, then you have GPS system, you have the vehicle system and then you have the auto link system as the main system which actually interacting with these external systems. So, we can actually identify the interaction between the external system and try to find out what kind of interface are needed between these external system, from the external system diagram. And based on this we can identify the external system requirements also and this also will become part of the document ORD.

(Refer Slide Time: 43:26)



4.0 Functional Architecture

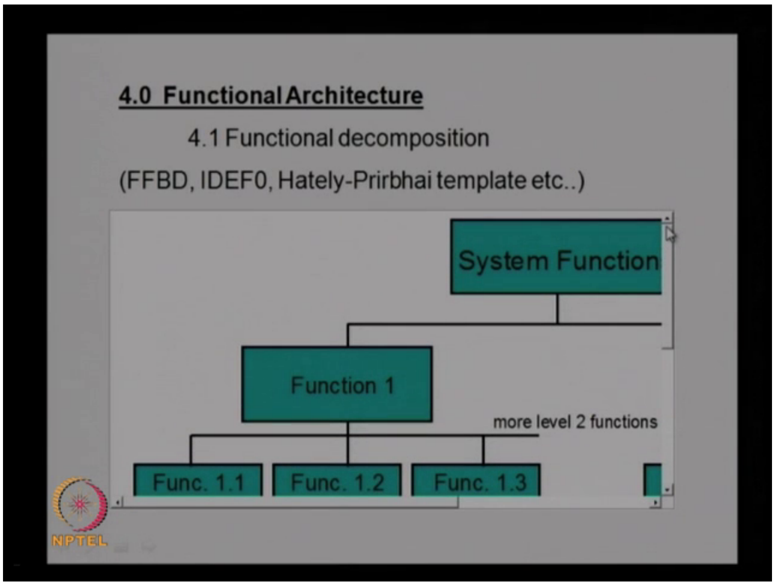
4.1 Functional decomposition

(FFBD, IDEF0, Hately-Prirbhai template etc..)

Shows the functions and hi

Same way you can based on these requirements, we can go for the functional decomposition.

(Refer Slide Time: 43:31)



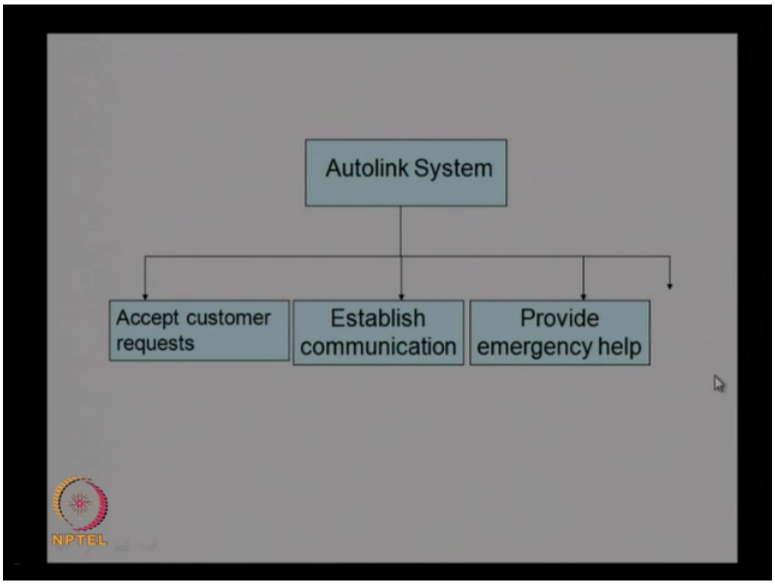
4.0 Functional Architecture

4.1 Functional decomposition

(FFBD, IDEF0, Hately-Prirbhai template etc..)

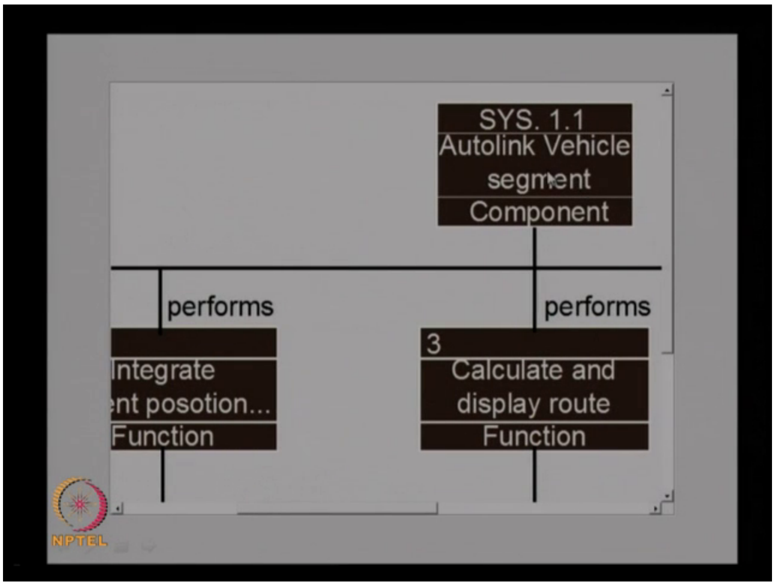
We discussed about two methods of IDEF0 and Hately-Prirbhai template; we can go for the functional decomposition based on this. So, we will get the top level function and all the sub functions and identify all the functional requirements based on this functional decomposition.

(Refer Slide Time: 43:48)



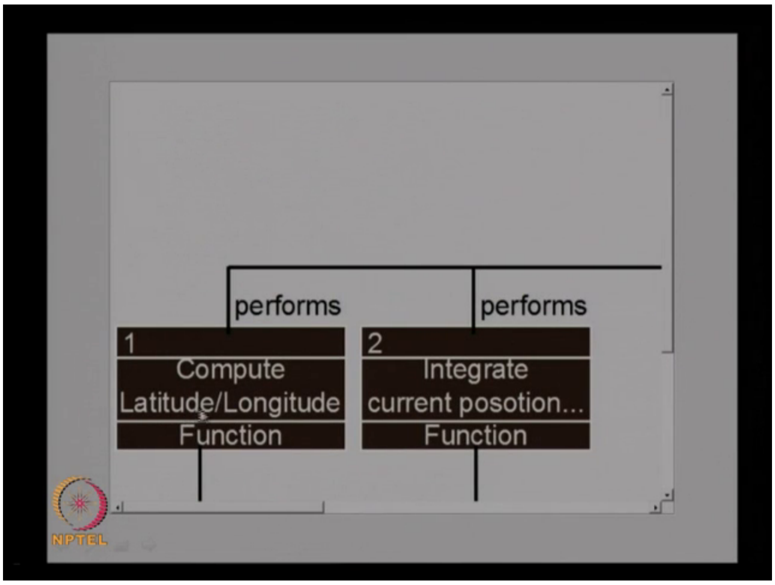
So, this is actually shows sub functions needed; so this is the main system, auto link system to provide the services. So, we have the sub functions as accept customer request, establish communication, provide emergency help and of course, you can have many functions like this. And here you can have many sub functions, you can decompose this into sub functions and till we reach to the lowest level of function, we will keep on decomposing the functions.

(Refer Slide Time: 44:14)



This actually shows diagram of the functional decomposition, so this is the auto link vehicle components function.

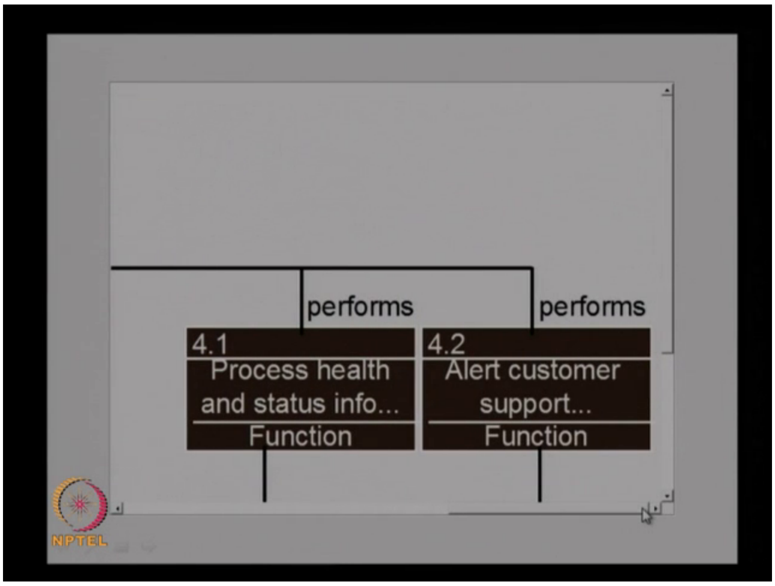
(Refer Slide Time: 44:25)



performs
performs
1
Compute Latitude/Longitude
Function
2
Integrate current posotion...
Function

Then the compute latitude, longitude is a function; then integrate the current position is a function, calculate and display route is a another function; then process, health and status information is another function.

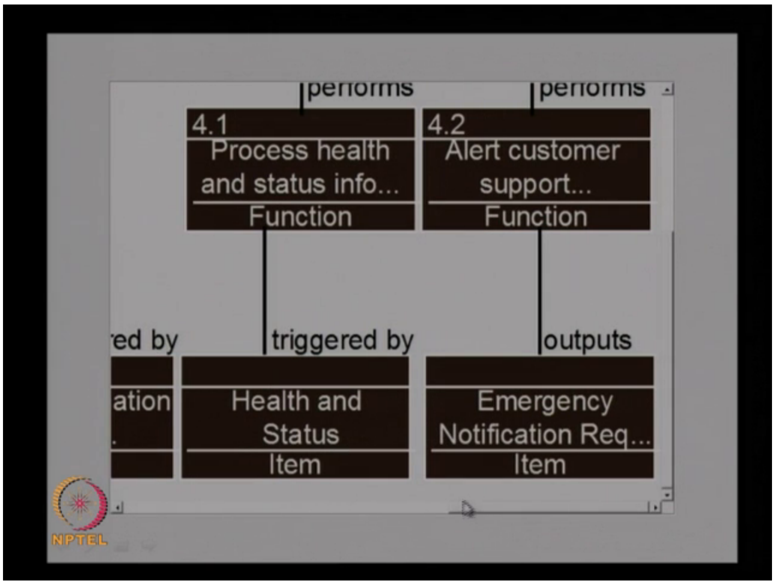
(Refer Slide Time: 44:38)



performs
performs
4.1
Process health and status info...
Function
4.2
Alert customer support...
Function

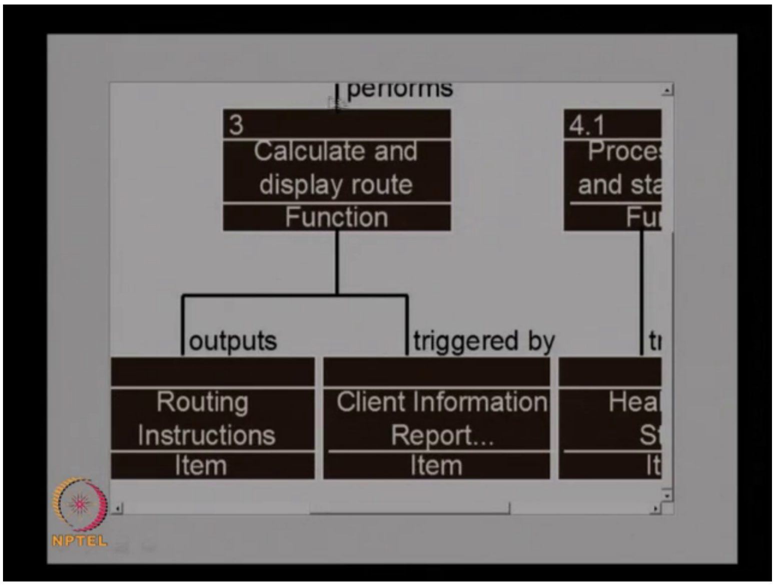
And alert customer support is another function. So, these are the function again this can an actually be identified.

(Refer Slide Time: 44:45)



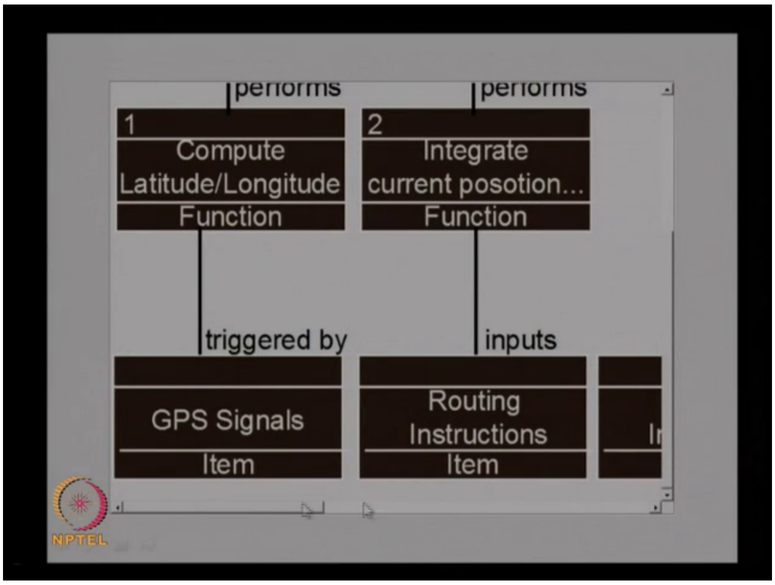
What kind of output is coming from here? And what is the requirement for these functions? Like you can see this particular function is triggered by the health and status of the system.

(Refer Slide Time: 44:56)



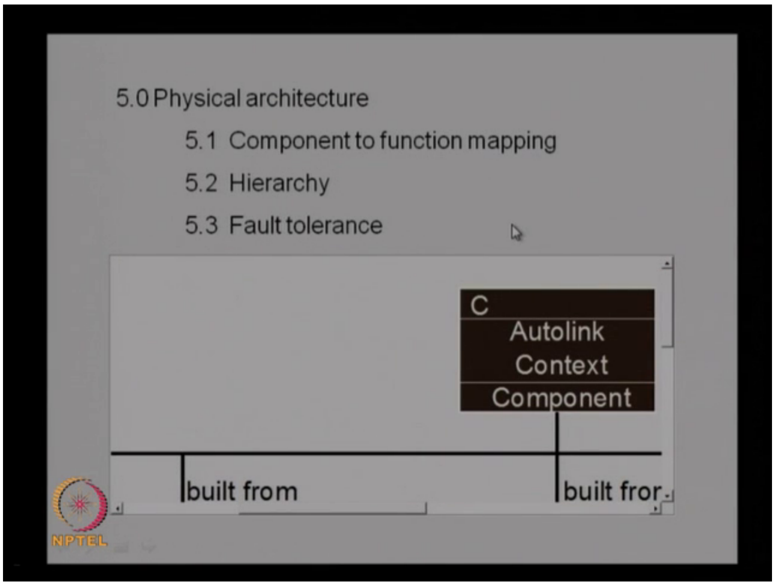
And similarly this particular function calculate and display route function is triggered by the client information report and then the routing instructions will be given from this particular function.

(Refer Slide Time: 45:07)



Similarly, you can see here also the inputs are routing instructions for this function and this particular function is triggered by the GPS signal. So, computation of latitude and longitude function are obtained from the GPS signals.

(Refer Slide Time: 45:21)



5.0 Physical architecture

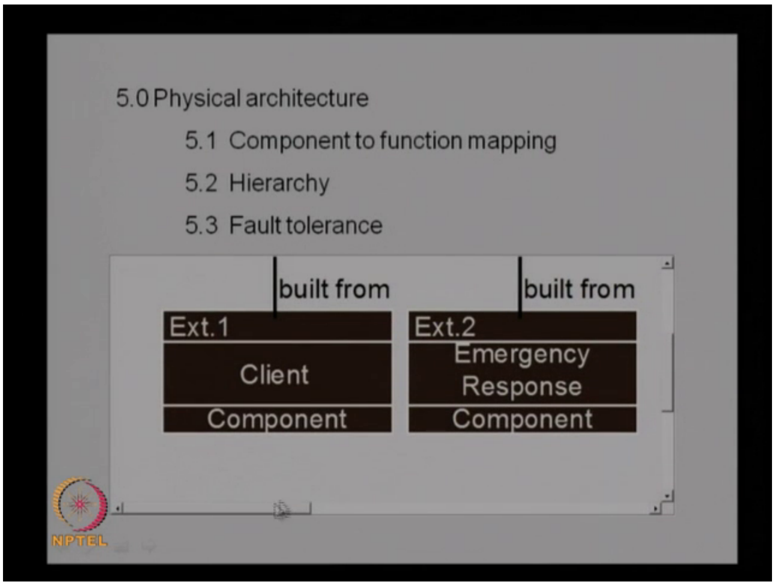
5.1 Component to function mapping

5.2 Hierarchy

5.3 Fault tolerance

And once you have these functional architecture, we can go for the physical architecture; basically we convert the functional diagram to physical architecture or a generic physical architecture. If we identify the components for various functions, so we can see here this is the main component here, the main component is the auto link system and then we have the subcomponents like the client.

(Refer Slide Time: 45:47)



5.0 Physical architecture

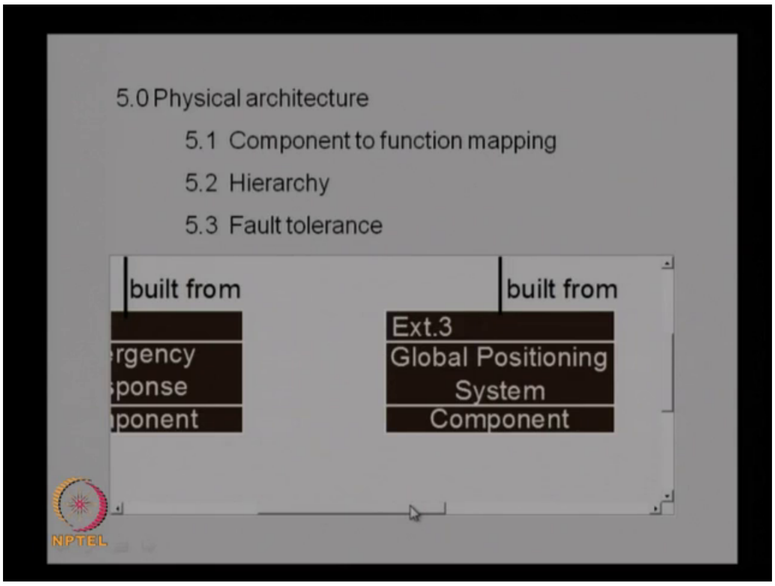
5.1 Component to function mapping

5.2 Hierarchy

5.3 Fault tolerance

Then the emergency response components.

(Refer Slide Time: 45:50)



5.0 Physical architecture

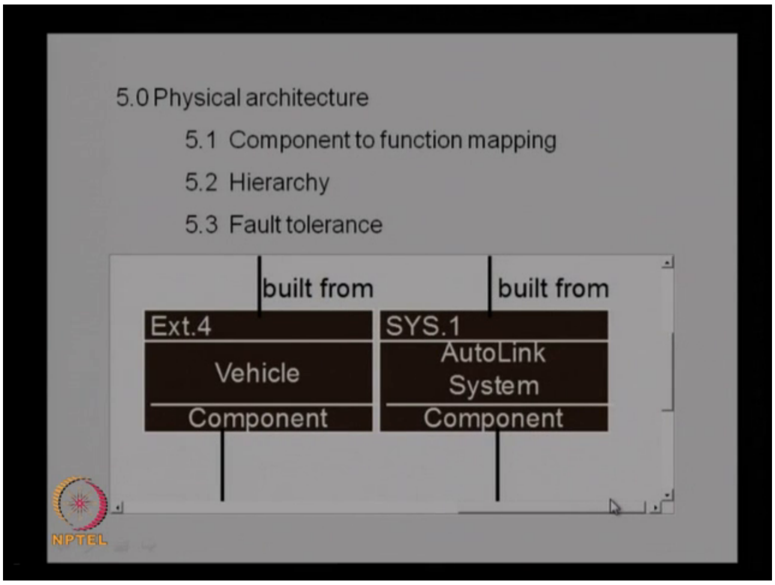
5.1 Component to function mapping

5.2 Hierarchy

5.3 Fault tolerance

Then we have the global positioning system components.

(Refer Slide Time: 45:53)



5.0 Physical architecture

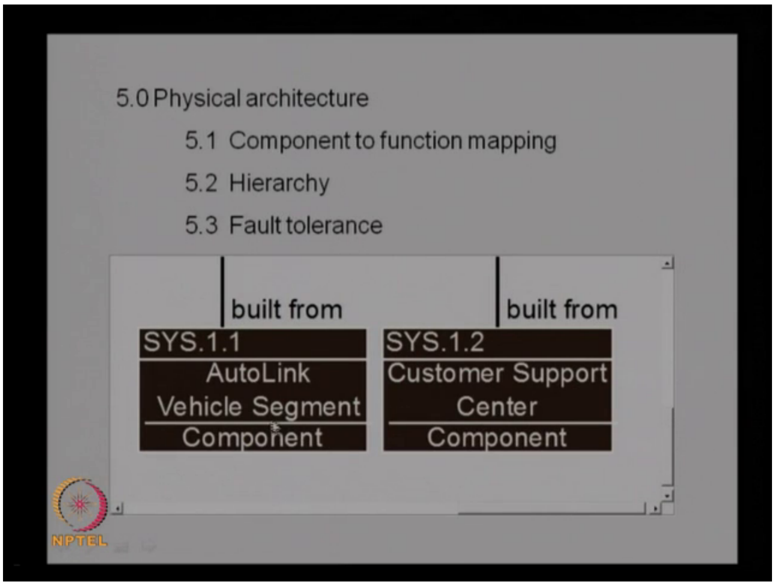
5.1 Component to function mapping

5.2 Hierarchy

5.3 Fault tolerance

The vehicle as well as the auto link system component and then this can actually be found out, this auto link system compound is built from the customer support centre then auto link vehicle segment is there.

(Refer Slide Time: 46:00)



5.0 Physical architecture

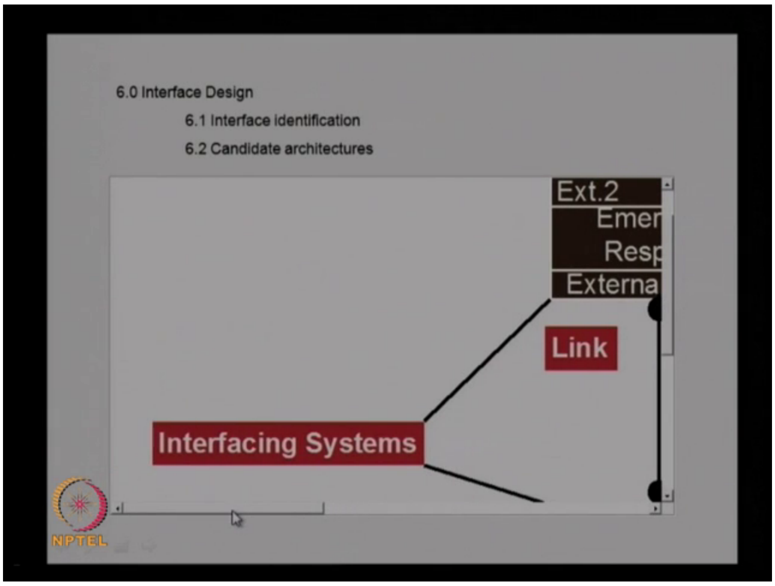
5.1 Component to function mapping

5.2 Hierarchy

5.3 Fault tolerance

So, like this you can actually decompose these functions into sub functions; sorry these components we can identify the sub assemblies and compounds which actually makes the physical architecture of the system. So, where basically we will do a component to function mapping, the hierarchy of the components and then we incorporate the fault tolerance components also.

(Refer Slide Time: 46:28)



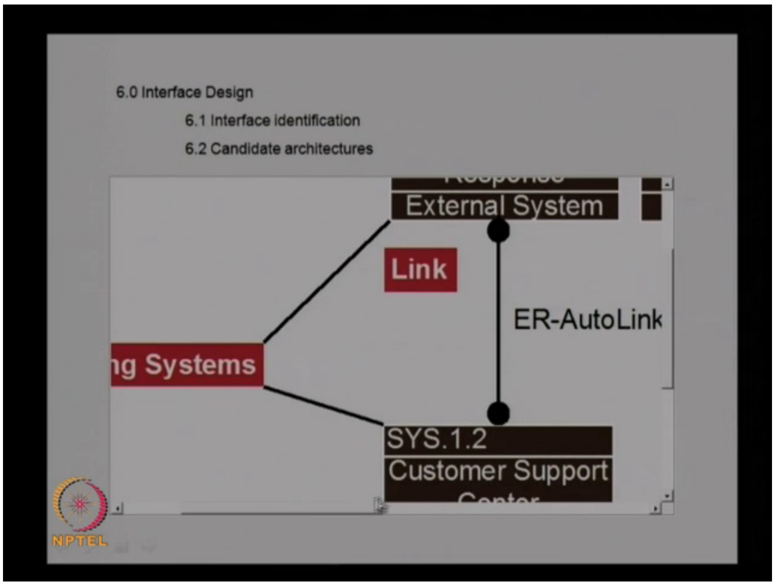
6.0 Interface Design

6.1 Interface identification

6.2 Candidate architectures

The final one is the interface design; so, we go for the interfacing systems, we tried to identify that interfaces in the required between the system and the external systems as well as the system and its sub systems.

(Refer Slide Time: 46:42)



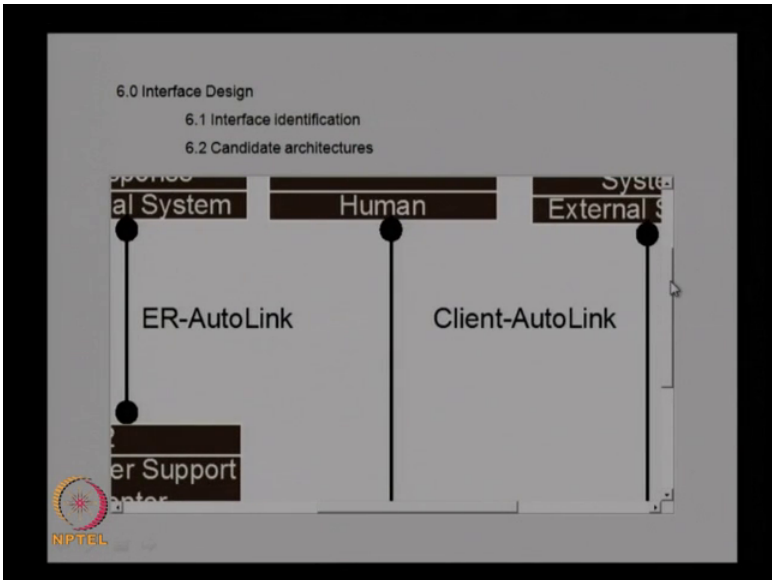
6.0 Interface Design

6.1 Interface identification

6.2 Candidate architectures

So, here we can see there are different interfaces between the emergency response and the customer support there is an interface, so it is a auto link interface.

(Refer Slide Time: 46:51)



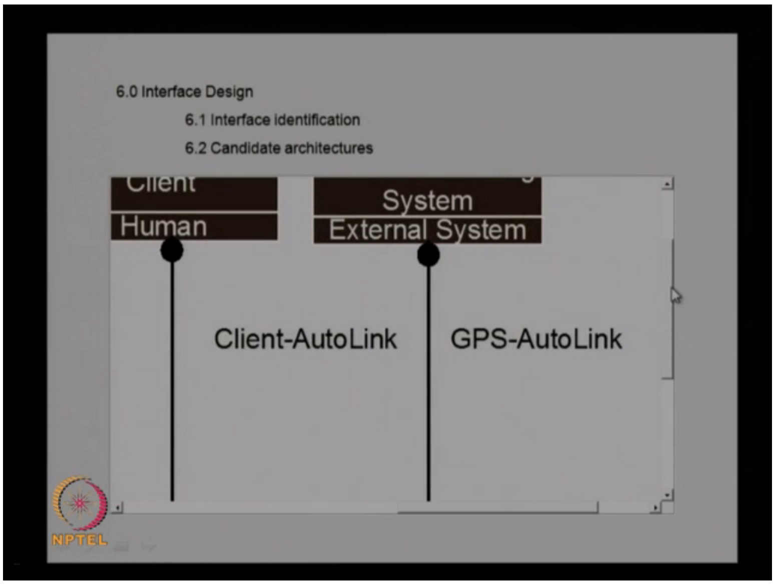
6.0 Interface Design

6.1 Interface identification

6.2 Candidate architectures

Then you have this human and the auto link vehicle segment interface.

(Refer Slide Time: 46:59)



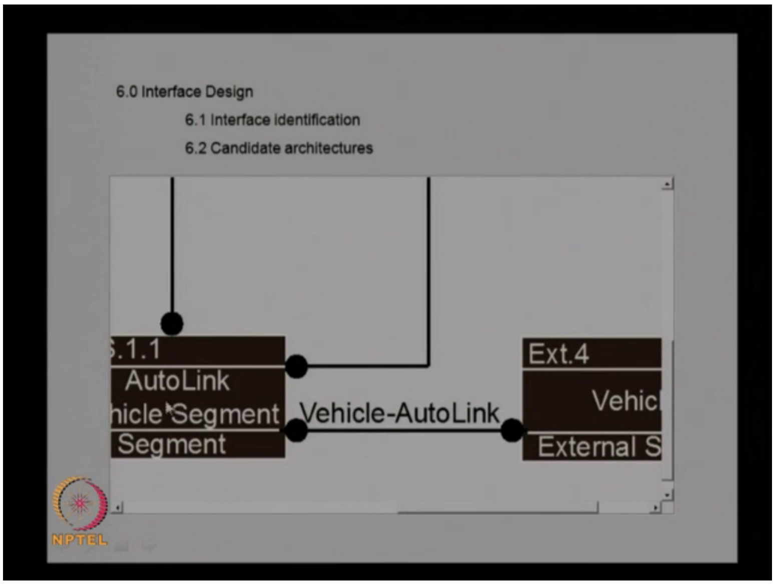
6.0 Interface Design

6.1 Interface identification

6.2 Candidate architectures

Then you have this system external system and the GPS auto link as an interface.

(Refer Slide Time: 47:03)



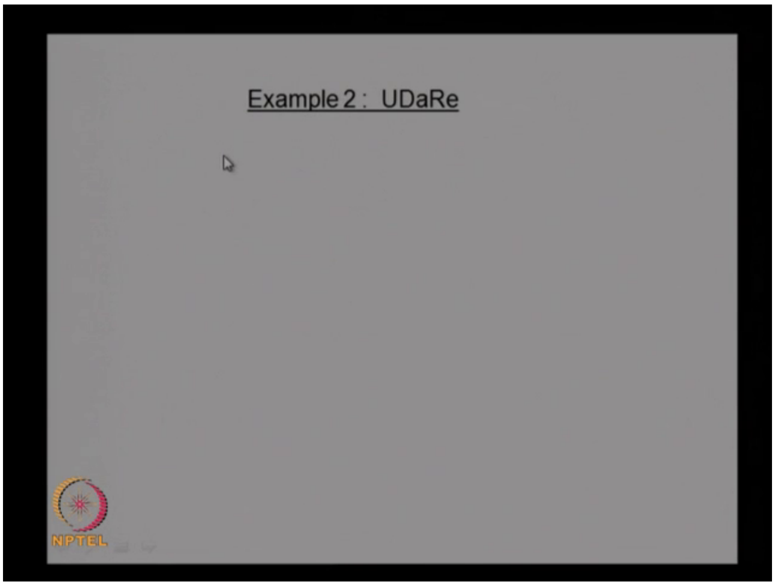
6.0 Interface Design

6.1 Interface identification

6.2 Candidate architectures

And this vehicle and the auto link vehicle segment has a vehicle auto link interface. So, like this we can identify many interfaces and then for interfaces, we can identify the candidate architectures for each interface; where we can go for wired interfaces or wireless interfaces that can be determined and incorporated in the design.

(Refer Slide Time: 47:23)



Example 2 : UDaRe

So, that is about the first case study, so we discussed about one case here what is the auto link system which is being used or which is being designed to help the drivers or passengers to navigation assistance, as well as for emergencies response. So, here we looked at the system; the system requirements or the functional aspects of the system.

Initially, we developed the mission statement for the system and then we are looked at the requirements of the system from the top level requirement, then specific requirements and then derived requirements, then we saw different scenarios of operation and based on the scenarios we are identified the input output requirements and we discussed about the external system requirements.

And once we identify all these requirements and the hierarchy of the requirements, we developed the originating requirement document and this originating requirement document functions as say basis for the design. And once we have this ORD, we will go for the functional decomposition, identify all the functions to be provided, you prepare a functional architecture and then based on the functional architecture, we will go for the physical architecture development. In the physical architecture development, we will try to make the components to functions and then identify all the components, the configuration items and then the sub assemblies and then prepare a physical architecture. And then we will go for the interface design, we will look at the interfaces requirements, then we identify the interface components and prepare a interface architecture also.

In the last one, once we have all the design we have to go for the testing and qualification of the system. So, that is how we develop the complete system; so, this was just one case study on auto link system. We will see a few more case studies in the next class before we go for the discussion of supplementary topics, so till you meet again goodbye to all of you.