Principles of Engineering System Design

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

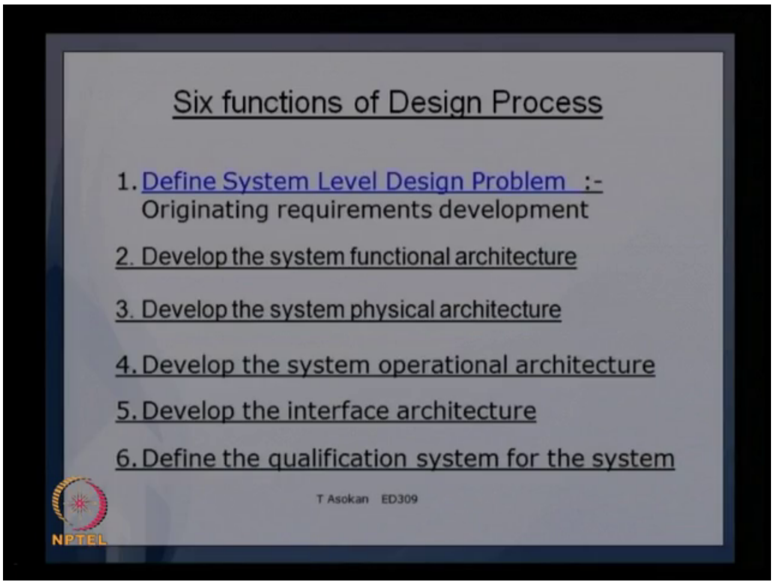
Lecture - 14
Operational Architecture Development
Part I

Dear friends, welcome back to one more lecture session on Introduction to System Design. Today we will discuss about the operational architecture development of engineering systems. Operational architecture also is known as the allocated architecture. So, we will be discussing about this particular allocation of a architectures for the system in this class.

In the previous lectures we discussed about the physical architecture development as well as the functional decomposition and then we saw that how the functions can be allocated to physical system and the physical architecture can be developed, where we can have a hierarchical architecture which can be used for identifying the systems and subsystems and assemblies and so on. But, that alone is not sufficient we need to look at the operational architecture especially when it comes to the internal architecture of the system, there are many things which actually could not be identified in the direct physical architecture development.

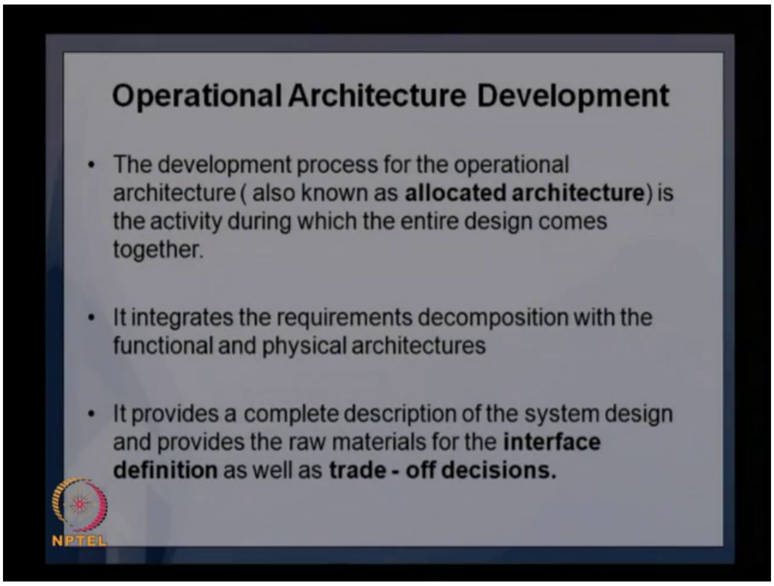
So, we will look at this part these aspects of the system development today.

(Refer Slide Time: 01:21)



As we can see, we were discussing about the 6 functions of the design process and then we completed the first three that is: the system level design problem, functional architecture and the physical architecture, the next one is system operational architecture. So, we will go to the details of this one.

(Refer Slide Time: 01:38)



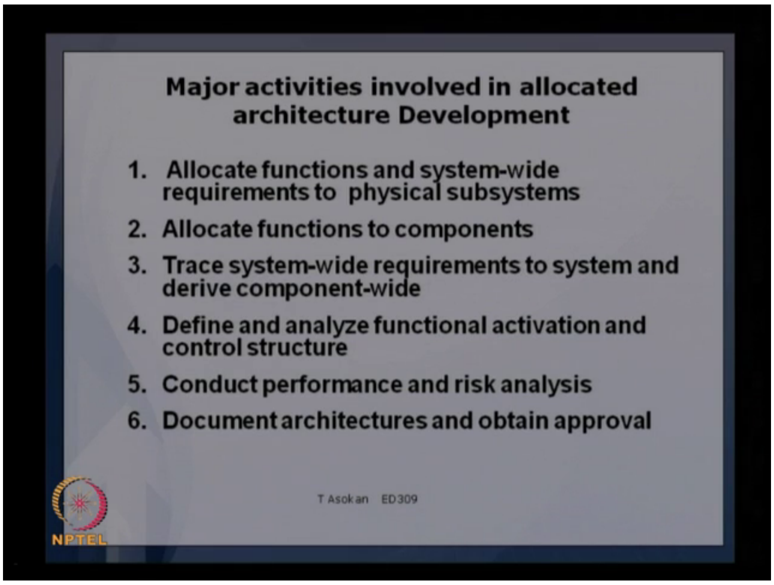
So, operational architecture is basically the development process for the operational architecture or also known as allocated architecture, this is the activity during which the entire designs comes together. So, this is the stage where we try to put everything together physical systems, the operational details and the internal requirements the technology in system wide requirements all those things are put together in this stage and that is the operational architecture development.

So, it actually integrates the requirements decomposition with the functional and physical architectures. So, we have the requirements decomposition and we have functional and physical architecture. So, we try to integrate these things together in this stage and it provides a complete description of the system design and provides the raw material for the interface definition as well as trade off decisions. So, as we know in the physical architecture development we have many systems and sub systems. And we need to have this system and sub system interface also along with the model.

So, we try to have the interface definition as well as the trade off decisions and the corresponding architecture also in the operational architecture development, in trade off decisions are basically whether we need to increase the cost or we reduce the cost only of performance or We try to have some other reliability or some other aspects trade off or some other performance aspects.

So, all these decisions also need to be taken and that also to be analysed and implemented in the system architecture. So, in the operational architecture or the allocated architecture, we will try to incorporate these features and then complete the system architecture or try to integrate the physical requirements, the functional architecture and physical architecture to get the systems operational architecture. So, that is the scope of this particular section.

(Refer Slide Time: 03:35)



So, the major activities involved in allocated architecture development are listed here. The first one is allocate functions and system wide requirements to physical sub systems. So, we discussed briefly about this in the allocation of functions and the functional architecture. So, in this stage we will actually allocate functions and system wide requirements to physical subsystems. So, there are system wide requirements like technology cost reliability manufacture ability or efficiency. So, these are the system wide requirements. So, we need to allocate these requirements also to components. So, which component will actually provide the required cost effectiveness or which component will increase the reliability of the system. So, that way we need to allocate the components to system wide requirements also. So, that is the first stage where we try to allocate functions to components based on.

Already identified the requirements as well as the system wide requirements and then based on this we allocate the functions to components. So, functions initially we identify the functions and the requirements and the functions and then we try to allocate them to the component. So, there are different methods of allocating the functions to components we will go through it later, then the trace system wide requirements to system and derive components wide requirements.

So, we have the system wide requirements, so we need to see how these system wide requirements are implemented in the architecture. So, we have to trace these system wide requirements to the system and its components. So, not only to the systems we can actually easily identify the system wide requirement to the system and we can say is provided, but we need to identify which component is actually providing these requirement and that is the tracing of system wide requirements to system and derived component wide traces.

The next one is define and analyse functional activation and control structure, this is basically to find out when a particular function should be activated or what are the input commands coming to activate the particular function, there are many sub functions or bottom level functions. So, each function need be activated at a particular stage. So, that is the activation signal coming to the function and we need to identify which is that signal which activate their particular function and what are the other control structures is needed, with what is the precedence of a particular function, whether this function need to be activated at a before the previous function or it that is to succeed a particular function So, that is a control structure.

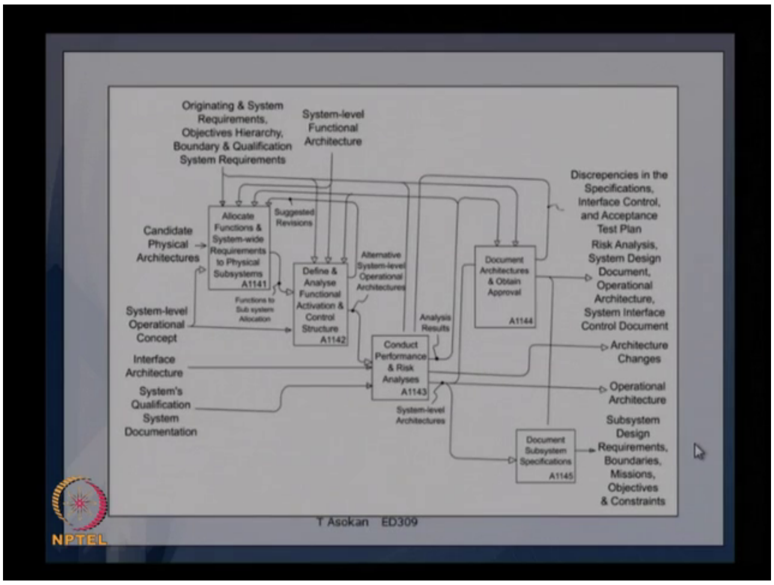
So, we will try to identify the activation functions or the signals as well as the control structure of the function in this stage. So, that also included as part of the architecture development and then conduct performance and risk analysis. So, what are the risks involved or which are the elements of higher risk or how do you distribute these risks or how do you identify the element which actually is of high risk in particular operation. So, all those need to be identified at structure development level itself. So, therefore, we do analysis to performance analysis to identify the risk involved in the subsystems as well as components and finally we document the architecture and obtain approval.

So, these are the major activities involved in allocated architecture developments. So, as I mentioned we will start with the allocation of functions to elements. Especially, we will look at the system wide requirements. And then try to identify the functions and then allocate that in to components and then we trace these requirements to component wide system or component wide.

So, we will try to ensure that there is traceability for these requirements and then functional activation and control structure will be identified and then risk analysis will be carried out. And finally we document the architectures and obtain approval. So, that is the different steps or different stages involved in developing the allocated architecture or the operational architecture of the system.

So, we will go through one by one and then see how actually we can do carry out these activities.

(Refer Slide Time: 07:44)



To represent these activities, the major activities of these development are shown in the IDEF0 diagram, I hope you are the now familiar with the IDEF0 diagram because we use it for the functional decomposition. So, if we represent the process using IDEF0 diagram you can see that these are the major functions. So, you can see here these are the functions 1141 11421 143 and 1145. So, this is 1144 and 1145. So, these are the 5 functions identified in the architecture development and then you can see what are the inputs and the outputs.

So, here the for the first one that is allotted functions and system wide requirements to physically sub systems, the inputs are originating and system requirements objectives hierarchy and qualification boundary and qualification requirements is one of the inputs and then system level functional architecture which we already developed is another inputs. Then the physical architecture the candidate physical architecture which we identified in the physical architecture development is another input. And then the other inputs are the system level operational concept which is coming from the initial stages.

And that will be going as an input and using all these inputs there will be an output coming as functions to sub system allocation.

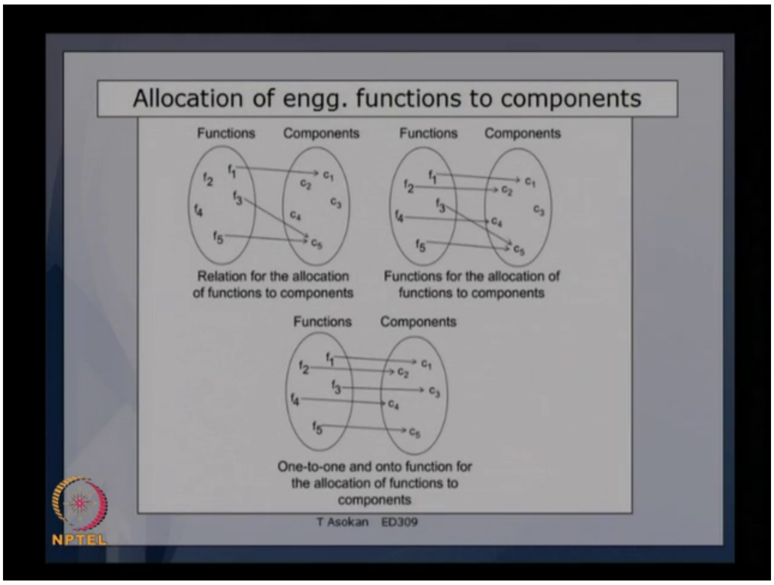
So, this is the output from this. So, you have the functions which are allocated to sub systems and define and analyse functional activation and control structure. So, this is as I mentioned you need to find out how are these functions need to be activated or what should be the signals which actually activated these functions that is to be identified here.

So, here actually the output will be an alternate system level operational architecture. So, we can actually get the multiple operational architecture alternative architectures because we can actually put in different order different format, so you will be having multiple architectures. So, that would be the output from this stage and then you can actually choose one of these architectures and then do a performance and risk analysis which is output will be the analysis results. And then that will going to the documentation for approval and that will be going as also document sub system specification. So, you can actually develop the subsystem specifications based on these analyses. So, output from these different stages.

Will we coming and 1 will be approved for send for approval and other 1 will be the used for sub system specification development. So, you can see these are the various outputs coming over here. So, this is the operational architecture document, this is the architecture changes whatever the changes taking place and this is the operational architecture.

Finally, you are getting all the allocated architecture and then the subsystem, design requirements, boundaries and objectives and constrains. All these will be developed or specified using this one, this is actually we represent the various stages using an IDEF0 diagram.

(Refer Slide Time: 10:42)



Let us go to the details of allocation of the engineering functions to components, this is what I mentioned we have to allocate the functions to components because, we have the functions identified in the functional architecture and we have the component identified in the physical architecture. So, we will try to map these functions to components, so what we try to do is to allocate the functions to components we do not do the components to function because of various reasons.

So, we can actually have mathematical relation type of allocation. Basically we have some relationship between these components and functions and allot the functions to components but then the problem here will be, because it is only a mathematical relationship some of the functions will be left out and some of the components also will be left out, so in order to avoid that we need to go for a mathematical function type of allocation. So we actually develop the mathematical functions relationship between the components and the functions and try to ensure that all these functions are related to components.

That is from these where we do a mathematical relationship, there is a possibility that to some of the functions will be left out. So, in this case the mathematical function method of allocation we ensure that this complete mapping is carried out over here. So, here we can see that some of the components are mapped to more than one function and some of the components are without any functions also. So, this actually is a problem because we have some component left out without any function, that it looks like the functional architecture was wrong or we identified wrongly the component to provide functions. So, this can actually be approached in a different way.

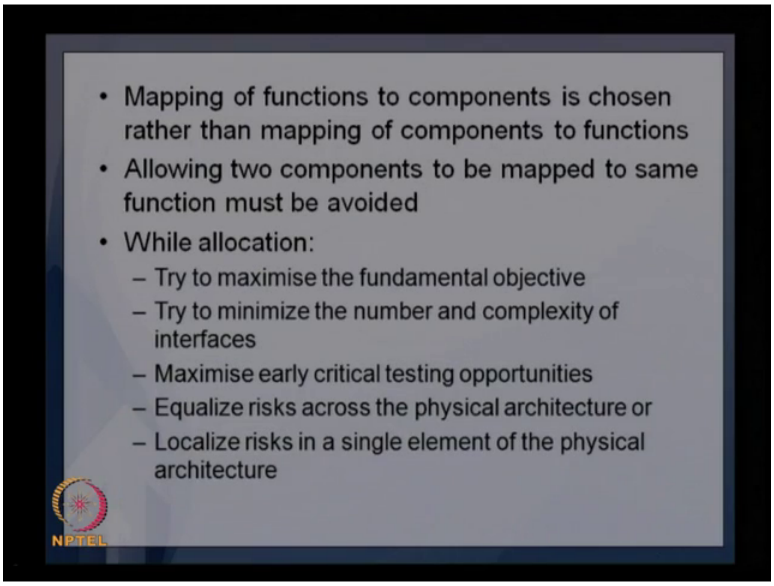
we can actually have a on to allotment that is you can actually have all the functions and all the components used in the system and we can allot these components also to some of the functions, then another problem here will be if you are try to allocation then that some of the components will be having multiple functions or some functions will be having done by multiple components.

So, like in this case it is not shown, but there is a possibility that saving function can be done by in this case, there is a saving function can be done by 1 component and in some cases 1 function shared between 2 components, suppose if the f 2 is map 2 C 2 and C 3. That means, these 2 components can actually do the function f 2. So, this also a problem because we do not know how to aggregate these function between these 2 components. So, that may actually create a problem at the later stage.

So, the best way is to do a 1 to 1 and 1 to 2 functions for allocation. So, we will develop a function which actually maps all the functions to components on a 1 to 1 and 1 to 2 basis that is 1 to 1 then only 1 component to 1 function. So, in this way we can ensure that all the functions are mapped to components and all the components have definite functions identified using this one. As in the previous case if you have some separate if same component having different functions or 1 function being shared by 2 component, the uncertainty will be there about the level of distribution of function or the components or if you have 2 components doing the same function, that actually shows that these 2 component one of the component is redundant.

So, in the normal allocation we do not have to have redundant components, unless it is for the error analysis or error reporting kind of fall tolerance system, otherwise we do not need to have 2 components doing the same function. So, all these can be avoided by going for a1 to 1 and 1 to 2 function allocation, which is normally employed for getting or mapping the functions to components. So, that is the way how the functions are allotted to components, this is actually a other view of the same thing for better clarity. So, you can see here this is the mathematical relation type allocation and this is the mathematical functions for allocation and the last one is 121 and 12 function for the allocation of the functions to components. So, this is the most preferred method where we allocate the functions on a 1 to 1 basis as well as 1 to 2 functions.

(Refer Slide Time: 14:51)



So, some of the important points about mapping the function, which is basically mapping the functions to components, is chosen rather than mapping of components to functions. So, functions to components not component to functions. So, this is what I explain we always do the functions to components because we do not want the many components doing the same function, that that will lead to redundancy. So, we will try to ensure that there is always a function to components mapping, rather than components to functions mapping.

So, the requirement is to provide the functions not to have the components in the systems. So, will look at the functions and then mapped to components if there are any redundant components we will try to remove them or we will try to identify separate function for the component or if 1 component is doing multiple functions. We will try to aggregate these functions into a single sub function and then provide the refined the architecture in the functional architecture in such a way that these functions can be aggregated to a sub functions.

So, that we do not need to go to further decomposition of that particular function because, there is already a components which provides the function. So, this way we can refine the functional architecture once we complete the mapping. Now allowing 2 components to be mapped to same function must be avoided. So, this I already explained so you cannot have 2 components to be mapped to same function.

So, this must be avoided because if you do this actually becomes a redundancy, 1 way it becomes a redundant system or the other it actually becomes difficult to aggregate these functions to components. So, which component will be doing the function or whether they are sharing the functions in a particular way, those things need to be identified over to the clearly stated.

So, in order to avoid that we will try in order to allow 2 components to be mapped to the same function; so these are the 2 important points to be noted when we do the allocation of functions and as I mentioned when we do the allocation basically there is a possibility that we can get alternate architectures because, when we have many ways of mapping the functions to 1 to 1 or 1 to 2 functions.

Then the possibilities are that we can have multiple allocation architecture can be developed and then when we develop this multiple architecture, we need to ensure that the choosing between these multiple architectures are done based on some objectives. So, if you have a particular descent objective then selection of this alternative structure should be done based on the descent objective. So, we will try to optimize the objective, suppose there an objective that the system performance should be improved in terms of a time of a service. Then the architecture which actually helps us to improve that particular objective to be chosen or in some places if we have a architecture or we have an objective to reduce the cost of operation.

So, if the that is an objective then we need to select now architecture, which will try to reduce the cost of operation rather than other performance parameters. So, like that we depending on the objective of the system development the architecture need to be chosen from the alternative architectures available. Similarly we need to try to minimise the number and complexity of interfaces also. So, whenever we have options for having different architectures, we should look at the number and complexity of interfaces.

So, whenever we have systems sub systems always there will be interfaces and we need to ensure that these interfaces are minimum, so of that complexity of interfaces also less. If you try to choose a complex interfaces, then the later stage when we develop the interface, we find a difficult to provide the required functional specifications or functional performance because a architecture is very complex.

So, whenever we have a possibility or we have an option to choose an architecture, we should try to minimise the number and complexity of interfaces. So, that is another point to be taken care of while choosing an architecture. Similarly maximise early critical testing opportunities whenever we have opportunities for early testing that should be accepted. So, instead of going for many complex systems, if we can have small systems tested or the testing procedure can be developed for small systems, where the particular group of functions are performed. So, that kind of architecture should be chosen rather than the possibility.

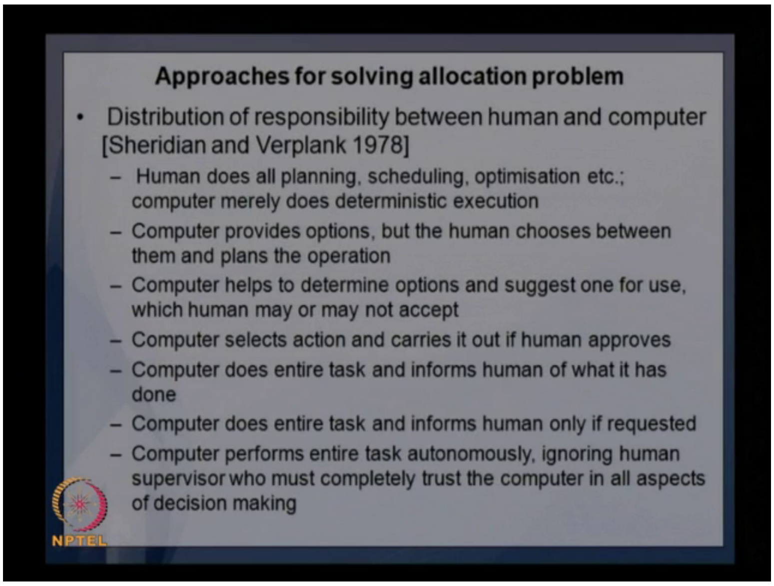
Having a testing only towards a end of the development that will be a problem because, we will not be able to the testing in before the actual implementation or before getting the complete a sub system. So, we should always try to maximise the early critical testing opportunities for the systems and sub systems and another important point is that equalize risks across the physical architecture, this is one of the requirements equalize risks across the physical architecture.

So, if the you identify some risks in the system, try to equalize or distribute it evenly across the physical architecture instead of making 1 system very highly risky, one particular sub system risky or some particular component very risky, try to equalize them the risk across the physical architecture or Localized risks in a single element of the physical architecture.

So, you can either distribute it equally or you can have a localized risk in a single element. So, we can actually identify the risk in those single elements. And then see whether that can be risk can be localized to that particular element so that we can ensure that either can be at that point only. These are the different points to be noted or to be taken care of while allocating the architecture or selecting the allocated architecture. So, since we can have multiple architectures or we have multiple allocated architectures, we need to chose one of the architectures and when choosing this architecture the focus should be based on these 5 points.

That is you try to optimize the desired objective, you try to have simple interfaces or less number of interfaces, try to reduce the risks by equally giving the allotting the risk to different components or we actually localize the risk in one component and then choose that architecture for a particular system, again it depends on the final objectives of the design as well as the application and all other factors. So, the selection of the architecture is clearly based on the design teams intuition, as well as their knowledge about the system and its operation, but these are the points to be noted and to be taken care of while choosing a particular architecture for the system.

(Refer Slide Time: 21:32)



And there are different approaches for solving this allocation problem because we have different ways of allocating the functions, especially when it comes to allocating work between the human and the computer or human and the machine we need to ensure that there are some standards followed in allocating of these function.

So, sometimes we can actually think of a completely autonomous system or we can think of a system where the minimal autonomy is there and most of the works are done by the human that is most of the systems are human cantered and human operated systems, we need to ensure or we need to find out what is the best possible way to allocate these functions between human beings and the machines or computers. So, there are some standard procedures and standard principles developed by various researchers. So, some of these principles can be employed, in order to ensure that we have some standards in following the architecture or allocating the functions to the components.

The reason why we need this is to ensure that, we understand the importance of machines as well as components computers as well as the human in the system and the depending on the objectives of the design, we can allocate these functions to machines or computers rather that depending on the ability of a system or the computer or a machine we will look at the system objectives and then allocate the function. The computer may be capable of doing many things, but whether you really need it for the system or not to be decided by the design team. So, that actually comes from the understating of the system or performance objectives as well as the development objectives, based on that the designers can actually decide how to allocate these functions to machines or computers.

So, some of the principles used for this one is basically known as distribution of responsibility between human and computer. So, that is the first this was actually propped by Sheridan and verplank in 1978, of course things have changed a lot during the last 40 years or so, but we can see that this actually gives a very general idea of how do we now the different ways in which we can allocate the function that is depending fully on the human being, for most of the task to the level were complete autonomy is given to the computers to carry out the task. So, if you look at the top level you can see human does all planning, scheduling, optimization and other tasks and computer merely does deterministic execution.

So, it will actually do some calculation inside the system, it will not do any other work it has most of the planning a scheduling optimization and all other things are done by the human beings only and computer is just involve used as a calculation equipment, where it is do some deterministic execution of some of the task and just give the results without really making any decisions or any planning all those planning the scheduling and all done by the human operators.

So, that is one way of allocating the functions the other one is computer provides options, but the human chooses between them and plans to operation, the other one can be we can find out options what are the possible options using a computer. If you want to arrive at a particular stage or even you want to move at a particular speed or particular location, you have to reach a particular location in a particular time. Then you can actually ask the computer to do some calculations and find out the different options.

So, if you would take a different path or take a different speed or you take a different attitude; so these are the different options the computer can provide you and based on these options, the operator can take a decision which 1 to be chosen; here the computer will take part only in giving you the options not actually choosing an option. So, that is the another level of implementation or another level of using computer for the functions, then next one is computer helps to determine options and suggest 1 for use which human may or may not accept. So, this is slightly a higher level where it not only determined the options, but it will actually suggest an option also.

It will tell this is one of the best options at this stage and whether to use it or not is actually completely left to the operator. So, here again the autonomy of computers slightly increases from the previous one, it is asked to give options as well as choose the best one out of the options given. And then the operator can actually decide whether to use it or not. The next one is computer selects action and carries it out if the human approves. So, here the actions are selected by the computer and it carries out the operation if the human operator approves it. So, giving more autonomy the computer has the autonomy to identify the possible actions and suggest the options and choose the options and it will execute it only if the computer if the operator says yes.

That is if the operator accept the options and the action to be taken then the computer itself will do the carry out action. So, here it is much more autonomy for the computer, it actually identifies the tasks to be done and the options and chooses the option and carries out the task. The next one is computer does entire task and informs human only if requested. So, this again top level autonomy. So, the computer does most of the work it actually identifies all the options it will identify the work to be done and it carries out and it informs only if asked by the human operator.

Well, if the human operator is not interested to know what is happening or what decision has taken. So, it will not even tell the operator it keep going on with its actions and probably the highest level is the computer performs entire task autonomously ignoring human supervisor, who must completely trust the computer in all aspects of decision making that is the fully autonomous system.

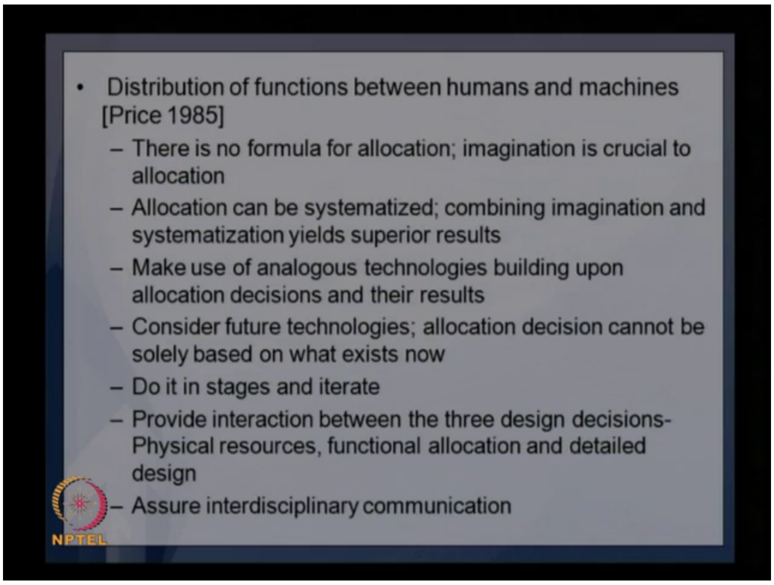
You can actually look at some of the latest system, if you look at an autonomous robot or if you look at the fly by wire system in the aeroplanes. So, these are all actually examples of how we actually allocate the functions to human and operator. So, in the fly by wire system most of the work can be actually carried out by the computer, computer can find out all the parameters needed, it can be identified the required authorised required direction changes, required attitude all those things can be calculated and it can actually control this and maintain that one. So, unless operate of the pilot wants to know this information will be available to the operator.

Otherwise it will be actually monitored continuously and maintain by the computer system itself, but in the case of an autonomous robots if you take an a completely autonomous robots we actually the operator actually plans everything and gives the task to the computer, the computer is do all the tasks it will do get all the information from different sources, it will help to find out the tasks involved to with objectives. And finally provide the output execute the task based on the information it provided with and there is no way the operator can interface or interact with the robot at this stage and it will be completely left to the robot to execute the task. So, that is last operator has to depend on the computer in all aspects of decision making.

So, these are the various levels of application or allocation of functions to computer. So, this was actually proposed by Sheridan a long ago and you can see that these various levels of this kind of allocations are implemented in various systems. So, depending on the requirement of the systems, it is possible for the designer to actually chose any 1 of these and implement that as a objective and then allocate the functions accordingly.

Again the final allocation depends on the designer, because he knows what should be the level of interaction, level of allocation of functions to the computer. And accordingly he can allocate the functions, so that is one way of doing it and another method or another principle is basically for distribution of functions between human and machines.

(Refer Slide Time: 29:33)



So, here it is not the computer but basically there are different kinds of machines, automatic machines, semi automatic, fully controlled by computer systems. So, how do I allocate the functions between human and machines in this case and this was proposed by price in 1985 and these are basically general principles again it will not get tell you how would how to do it actually, but basically tells that what are the binding principles on which the allocation should be based. First one says that there is no formula for allocation imagination is crucial to allocation. So, you cannot really have a formula to allocate the functions, so it actually depends on the creativity and the imagination of the designer, which is very crucial in allocating the functions to machines and allocation can be systematized, combining imagination and as systematization yields superior results.

So, is second one says it can be systematize. So, we can have a systemic way of doing it, basically looking at the requirements looking at the functional capabilities of the machine and looking at the risks involved in different task we can do a systematic allotment, but if you combine the systematic allotment with the creativity or imagination we can actually get superior results in allocation of functions, another way of doing it is make use of analogist technologies building upon allocation decisions and their results. So, analogist technologies are the technologies existing in different other places or different other systems. So, you look at those technologies or the system analogist to the present 1 and build upon that 1 for allocation of functions and look at the decisions and the results in those analogist technologies, use them for allocating the functions to machine.

So, some of the thing already existing there are some machines which actually do some kind of tasks. And there are some kinds of allocation of functions to machines, existing in other fields when you look at that one. And then see what kind of decisions are made for the allocation and what are the outcomes and what are the problems faced by them and based on that we can actually have a systematic way of allotting function. So, this is basically looking at the existing analogist systems and then allocating function accordingly and another point to be taken care of the future technologies.

So, we need to make sure that whatever the allocations we do it actually caters to the future technologies also. So, the allocation decision cannot be solely based on what exists now. So, we need to look at the future and what kinds of technologies exist or what are the possibilities of changes in technology. And accordingly, we need to allocate the functions and ensure that the future technologies can be easily adopted or when the technology improves some of the functions previously done by human or some other system can be moved to the machine. So, that is the future technologies there are lot of developments taking place in the technology sector. So, there is a possibility that the work being done by human today can be done by a machine tomorrow.

So, we need to have the functional structure in such a way that this task can be or the functions can be easily mapped to the machine at a later stage, that is why we should look at the future technologies and allocate functions accordingly and again do it in stages and iterate may times. So, do not do it the complete allocation until 1 go do it in stages allocate some of the functions and then iterate it many times and then see; what are the possibilities and then complete the structures.

So, do not do it in a single go do it in stages and iterate and that actually helps us to identify the possibilities of failures or possibilities of interchanging components or the functions. If it we do it in stages if you do it in a single go then you may lose the focus and then probably you do not get really get the alternative architectures what we need and providing interactions between the 3 design decisions.

So, this is very important as you know there is the physical resources, function allocation and detailed design are the 3 design decisions, when we make a decision or we design we will need to look at the physical resources available, we need to look at the function allocation what are the functions to be provided and the physical resources and then detailed design of component. So, once we have these components we need to go for and identify the components we need to go for the detailed design of these components.

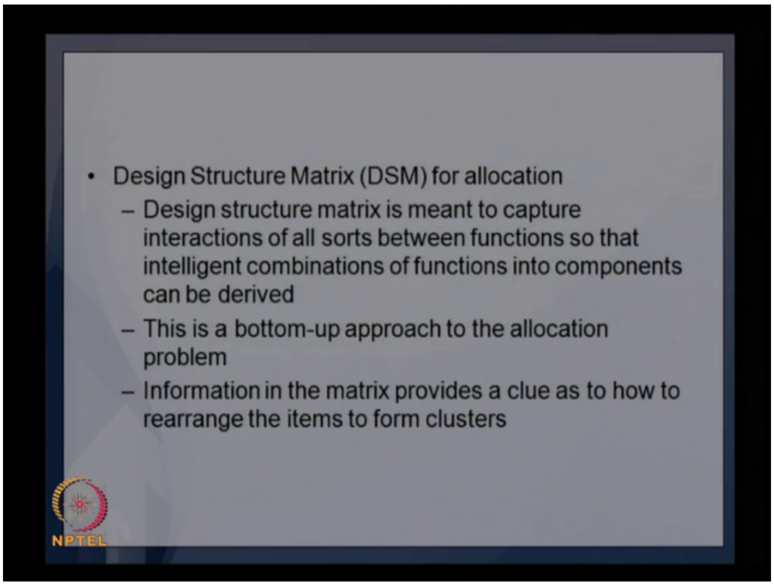
So, we need to provide sufficient interaction between these 3 stages. So, whatever may be the allocation we are doing, we should look at the physical resources available and the detailed designs strategies or designed requirements and we should provide sufficient interaction between all these three. In order to make sure that we have a good system or we have a perfect system at the end of all these 3 design decisions.

That is the providing interaction between the design stages. And finally assures the interdisciplinary communication again in allocation there will be multiple disciplines multiple domains involved in the system. So, the communication between these disciplines or the domains becomes very important. So, we need to make sure that the function allocation actually takes care of the interdisciplinary communication and ensure that there are ways to communicate between these domains or there is a common vocabulary or a common strategy for communication between these domains.

So, these are the different principles on which the allocation of functions between human and machines is to be based upon. So, in the previous one we discussed about the allocation of functions between human and the computer and that was mainly based on the autonomy requirement and this one actually discuss about the principles, how we actually allocate functions between human and the machines. So, the important points to be noted here are futuristic outlook that is whenever there is a technology improvement, we should be able to get the technology properly.

We should be able to adopt to these technology the other one is the interaction between different stages of design that is the different elements of design that is physical elements available, the allocation of functions and the component design that is the detailed design of components. So, these 3 should be there should be sufficient interaction between them and there should be a proper communication architecture or communication interface between different domains in the system. So, where this mechanical domain or in electronics domain or a communication domain elements, we should identify the interdisciplinary communication requirements. And accordingly provide the system architecture. So, these are the points to be taken care while allocating functions.

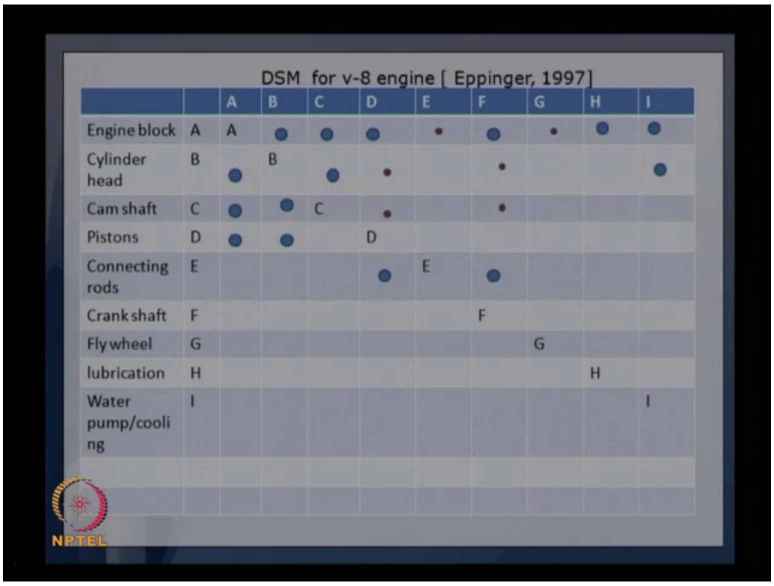
(Refer Slide Time: 36:18)



Another method of allocation of function is known as design structure matrix for allocation. So, design structure matrix is basically we do not again allocate the functions directly to the components, but actually it helps us to identify the interaction between different components and then to identify some clusters of components or clusters of functions which can actually be use at the future stage to optimize the design activities or improve the design activities. So, this design structure matrix is meant to capture interactions of all sources between functions. So, that intelligent combination of functions into components can be derived. So, basically we are trying to have combinations of functions into components though we can have identify the functions and components and then see what kind of interactions are needed when during the design of these components and based on them we can form some clusters and this is a bottom up approach because, we start with the components and then try to identify the clusters.

That is why it is known as a bottom up approach through the allocation of problem and information in the matrix provides a clue as to how to rearrange the items to form clusters. So, here actually it provides a clue to rearrange the items to form clusters, I will take a very simple examples not a complete one, but a partial partially completed matrix to tell you how to use this 1 for allocation of functions or identifying the clusters .

(Refer Slide Time: 37:45)



So, look at this DSM there is the matrix you can see that these are the different components identified based on the functions and functional requirements, these are the components identified like for a engine that is v 8 engine this is can be this is from Eppinger 1997, if you can actually refer to these and then get the complete matrix.

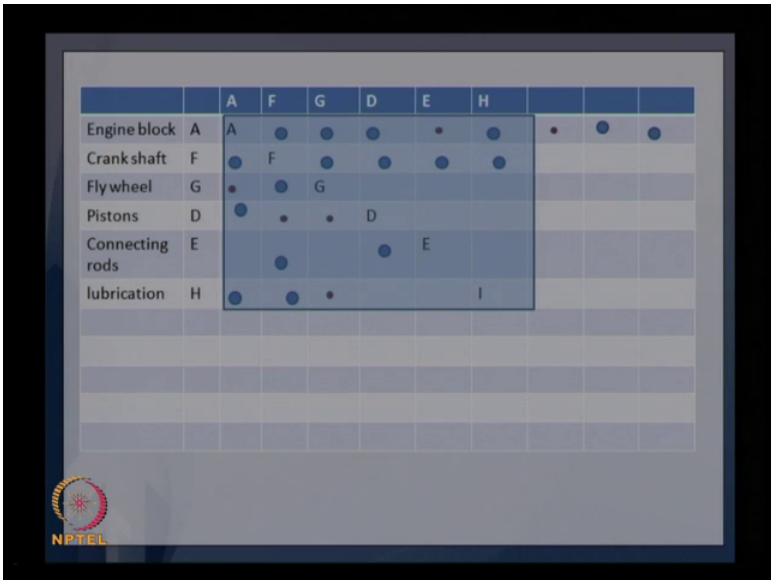
You can see here these are the different components, so you can as engine block cylinder head cam shaft piston connecting rod of course; there are many I just shown only few over here. So, how do we make the matrix is that we write down these components on the rows and then we right down the same thing A B C D E F G. So, this is A B C D E F G in the columns also and then the rows actually represent. So, which are the inputs required by this particular block, suppose engine block designer what are the inputs required by them from other sources those are actually marked here like circles over here dark circles with strong interaction. So, you need lot of information from B that is the cylinder head you need lot of information from C cam shaft you need the lot of information from d that the pistons and these interaction is not so strong that is why its shown as a very small circle.

So, connecting to rods you do not need to have that much of information for the design of engine block, when then like this you can have G is the fly wheel and lubrication. So, you can see that fly wheel information is not so important for the engine block, but lubrication is important. So, will have lot of interactions are taking place here. So, like this we can actually indentify the information required or the input required by this design team for designing the engine block and the vertical or the columns actually represents what are the outputs going from A to other design teams for example, if you take the engine block design team A.

Output from A will be going to the cylinder head the cam shaft and the pistons as very strong interaction; that means, lot of output from A is needed for B C d and others. So, similarly you can see that here the design team the C team the output from C will be used by these blocks. So, these design teams of course, this is not complete this is just an indicative of the interactions, but the interactions are represented by circles strong interaction or strong requirement of data is represented by a dark circle and the other 1 is represented by a small circle. Now looking at these interactions it will be possible for us to identify which are the blocks should come together, because they have lot of interaction lot of data exchanges taking place, they need to discuss a lot between them to get the design successful. So, we will try to cluster them together and make them as a larger group.

So, that is the idea of using DSM we look at these and then try to have a blocks along diagnose. So, wherever the interactions are more we will put them together and then make it as block and that is done the next one you can see here.

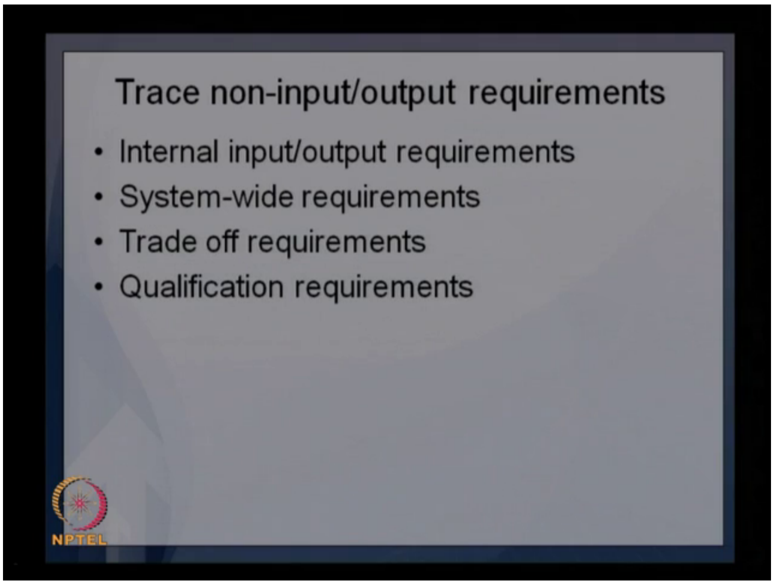
(Refer Slide Time: 40:53)



So, you can see here this engine block crank shaft fly wheel piston connecting rod and lubrication, they have lot of interaction between them and that is why this has been made as a single block. Similarly it will be identifying other blocks also along these lines based on the other components and their interaction and we will try to identify the clusters of design team who needs lot of interactions.

So, this also has helps to allocate the functions accordingly, because the interactions are more over here and therefore we can actually have them to cluster together to get the design effort in a better way or to focus design effort. So, this is the use of DSM in application of identifying the functions to be allocated or it will help us to allocate functions accordingly.

(Refer Slide Time: 41:38)



so that was about the allocation of functions and the different methods for allocation of functions and the principles to be used for allocation of functions and other things to be done in a important things to be carried out or important activities to be carried out in the allocation architecture is basically one is the trace non input output requirement. So, we have input output requirements which are basically traced using IDEF0 diagram, but there are other things to be done. Basically, the internal input output requirements system wide requirements trade off requirements and qualification requirement.

So, these need to be traced into the system when we develop the allocated architecture. And we will discuss these about the tracing of these non input output requirements in the next class, till we meet the next time.

Thank you and goodbye to all of you.