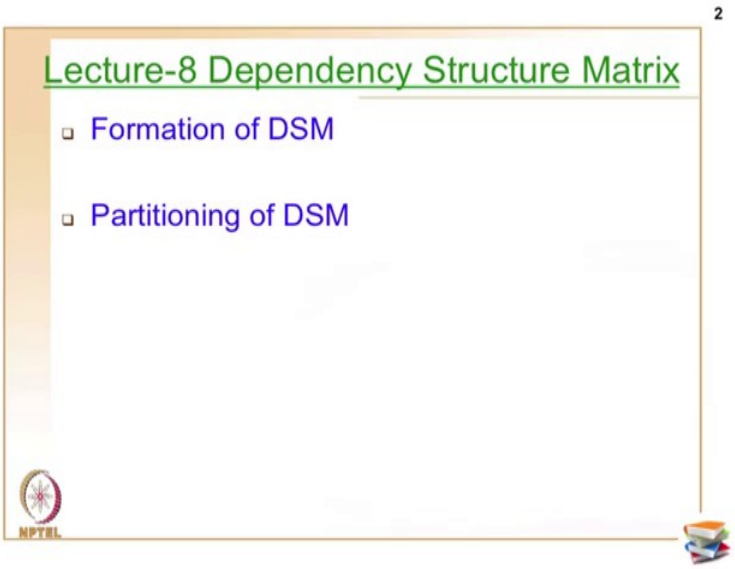
Scheduling Techniques in Projects
Dr. J. Uma Maheswari
Department of Civil Engineering
Indian Institution of Technology, Delhi

Lecture - 08
Dependency Structure Matrix – I

Title of this today's lecture is on Dependency Structure Matrix. In the last class, I have introduced and I gave you slight background on what is design management, what is information and so on. So, these with the help of this a technique called DSM, you will understand how to also go ahead in finding out scheduling aspects in DSM. So, let us move on the contents.

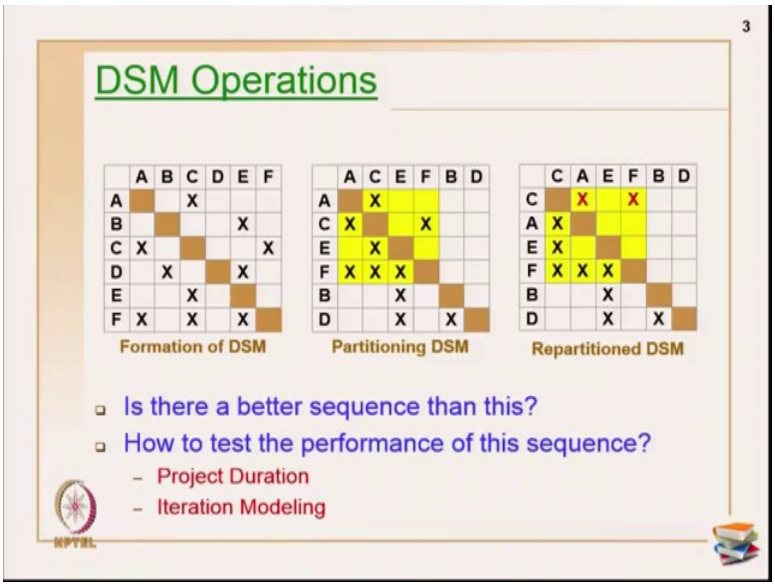
(Refer Slide Time: 00:50)



In the last I have also shown you what are the basic operations in DSM like formation of DSM, partitioning of DSM, then tearing process and iteration modeling.

So, in that first two operation basically the formation and partitioning is what we are going to see in today's lecture.

(Refer Slide Time: 01:03)



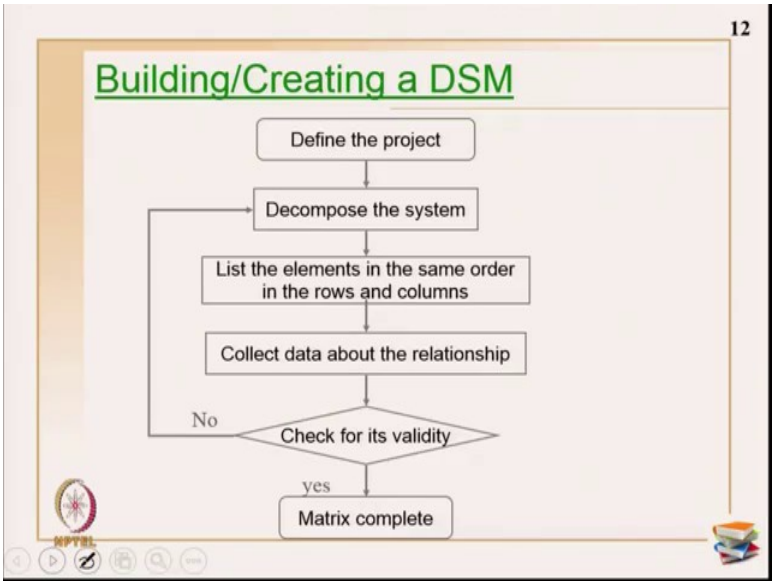
Just to brush up your mind. So, this is a formation of DSM, I have taken a same example as shown in the last class. This is a partitioning where in for this particular example, I am getting one block ok.

And then once you do the tearing process, in the sense identifying the tear marks or place for assuming then you get the tearing block like this. And once you finish, then you have to again do the repartitioning and you will see this block. So, primarily the three operations are here; this is operation number one: formation, this is operation number two which is a partitioning, the third one is tearing. Tearing always goes along with the repartitioning, and then the next step is primarily the iteration modeling. So, now, unless you stop with the three steps, you are only identifying the sequence outside the block, with the partitioning you are identifying the sequence outside the block. And you are also aware of where are the block formations ok.

In the tearing and repartitioning what you are going to do is, you are not touching outside the blocks you are only going to operate within the block. So, within the block where are the tear marks and what is my sequence within the block and what is my process of execution that is what you are going to see in tearing. Now the question that comes to your mind is, is there a better sequence than this; for example, C A E F and then B D. So, primarily with the C A E F is there another alternate sequence? So, primarily for factorial combination sequences are possible for the four activities within the block.

Now so, accordingly you have like 16 combinations are possible ok. So, one way is to go ahead in evaluating all the 16 combinations and then seeing what is a best duration or other way is primarily to do iteration modeling ok. So, how to test the performance of the sequence, the only one method is add duration along the diagonals; and then find out the duration for this project or for the block whatever it suits well. And then you have to find, what is the, how much of rework has happened, how much of waiting time was there, and what is happening, which activity is taking more time in the rework; so you can analyze with the help of iteration modeling, that is a phase number 4 or operation number 4 ok.

(Refer Slide Time: 03:30)



So, now as I told you, we are going to see with the formation of DSM. So, building or creating a DSM, in other words we call it as a formation of DSM. I gave a very simple steps, this is like it can go through for any way with which you are creating a DSM ok; which also I will little bit I will explain later. So, first step is to define the scope of the project ok. So, for example whatever project you want to take, how many our disciplines are there, how many our components are there, you choose accordingly, and we define what is the scope of the present project.

Then you decompose the entire project into various elements as possible. So, you can stop at level 1, you can stop at level 2. So, the common method of decomposition we have seen in the earlier classes is WBS, you can go through the same WBS methodology

in order to decompose a system. And you can stop at level 1 2 3 wherever you want, you can still do the same and you can build the DSM; I will show you that also in today's class with various levels of the DSM formation.

Then identify the list of elements; elements is a common name which is given for either you want to do component, teams, activities, parameters, whatever. So, when I am decomposing a project, I may land up in so many different sub-levels which we have shown in the last class. So, it can be components, teams, activities, deliverables, parameters etc, so you can stop at any anyone place and accordingly you can create that particular type of a DSM.

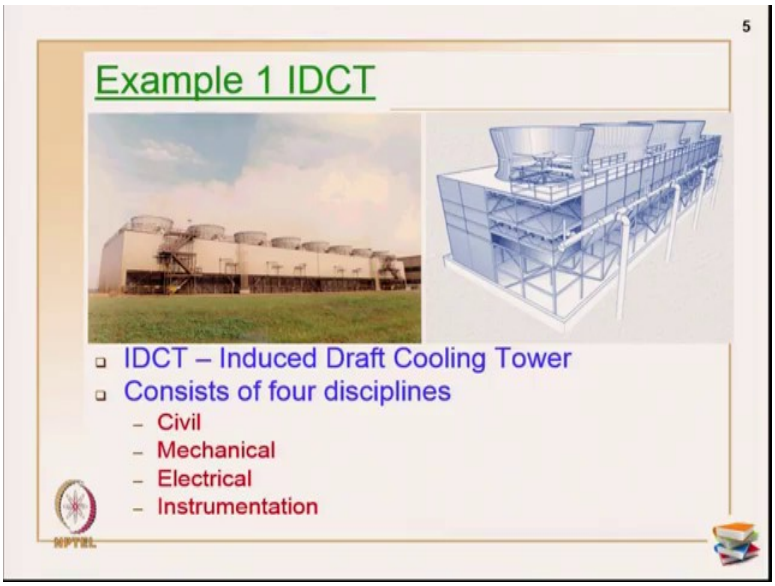
So, if you stop at activity DSM activity level, you can form an activity DSM; if you stop at component level, you can form a component DSM. So, there are so many types of DSMs also available. So, list the elements in the same order in the rows and columns. So, accordingly what you have to do is, you have to identify the elements; the elements is generic name given for components, teams, parameters and so on. So, if you want to do out activities; then list out all the activities in the same order in rows and columns, so that you are getting the skeleton DSM.

Skeleton DSM is a blank DSM with just the rows and columns identified and no relationships are identified ok, that is primarily called skeleton structure. Then collect data on the relationship, once you have listed out all the rows and columns; the next step is finding out the relationship between those identified rows and columns ok. For that we are suggesting a method with which we have worked out; and that you may follow or there are several other methods with which you can also build or define the relationships ok.

Then check for the validity, once you have done all the X marks it does not mean it is fully complete, go through the experts and see whether you there is still any unidentified relationship or whether there is a wrong identified relationship, something is missing and so on; that you have to cross check.

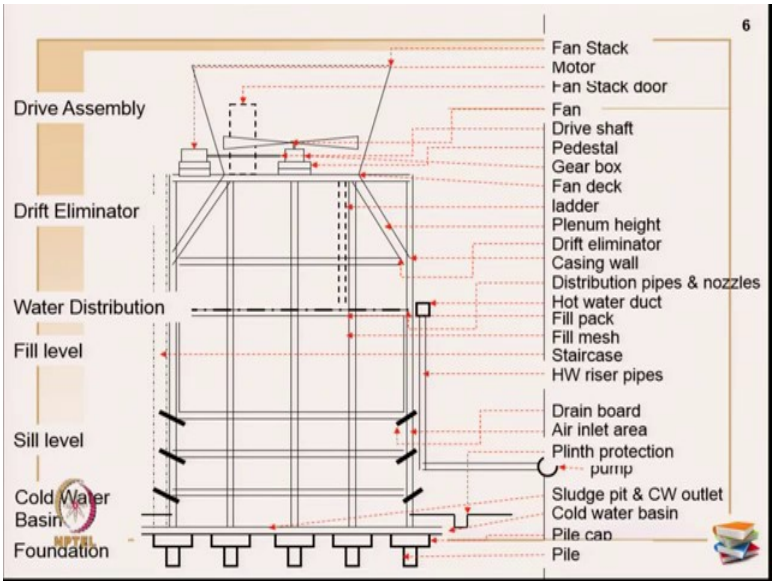
And once you are very clear that you have done everything perfectly fine, then you can really go with assurance that the matrix formation is fully complete. So, this is a very basic steps with which you can create a DSM. Now let us go through the steps in detail with an example.

(Refer Slide Time: 07:03)



The example, first example which I have taken here is IDCT, which is induced draft cooling tower ok. This example it consist of four disciplines such as; civil, mechanical, electrical and instrumentation. So, there are four disciplines, for the particular scope of this IDCT project ok. So, that is the reason why I have chosen this.

(Refer Slide Time: 07:19)

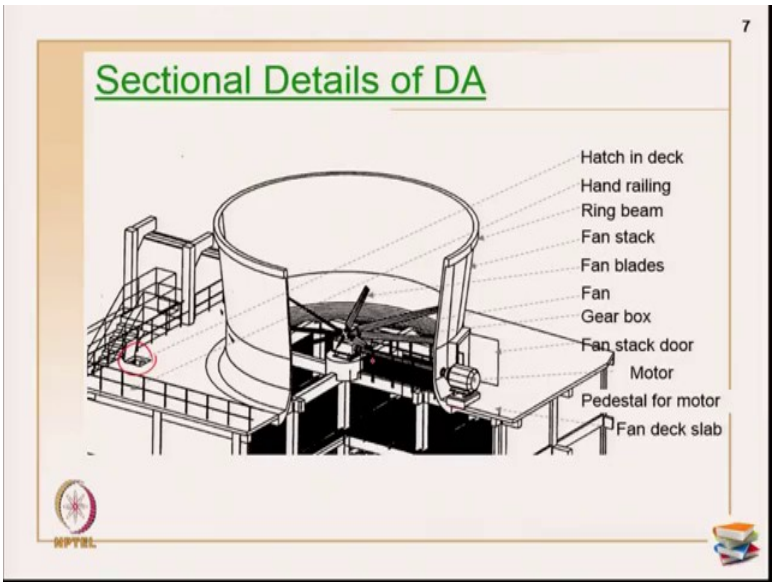


Now, the cross-sectional details of this IDCT is given here or the induced draft cooling tower is given here. I think many of you know what is a cooling tower? It is primarily a part of a power plant project in order to cool the water which comes from the turbines

ok. So, now, what are the levels? So, primarily it has several levels one is; drive assembly, drift eliminator, water distribution, fill level, sill level, cold water, basin, foundations and so on. So, the water which comes, water is generally coming through the hot water basins and it moves in through the drift eliminators and the water distribution lids ok.

What happens here is the fans and the drift eliminator prevent the water particles from going outside ok. So, this is primarily the hot water basin with which the water is falling in and in the fill level there are the cooling medium ok, which cools the water. And once the water is falling on at very large level, naturally the water gets contact in touch with the air and the water gets cooled; and then it starts going into the cold-water basin where the cold water is collected again for recirculation of the water process ok. This is primarily the mechanism with which the IDCT works.

(Refer Slide Time: 08:46)

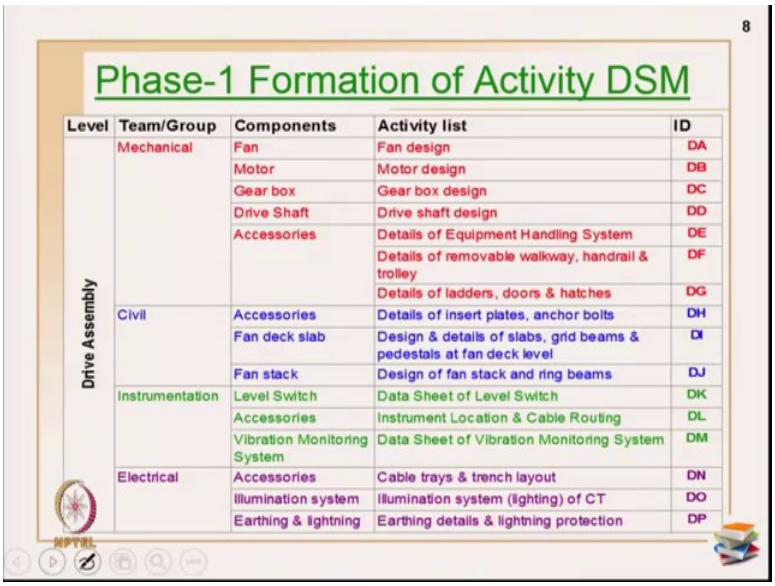


Now for the particular example in formation of a DSM, we are going to see only the Drive Assembly level, which is the topmost level ok. So, little more zoom in view on the drive assembly level, to show the sectional details. If you see here, so this is primarily the, let me use my pen ok. So, this is primarily the hatch in the deck for all the operations to happen and these are all the handrails are shown here, this is a ring beam, this is a fan stack; fan stack is primarily in the cover to the fans ok, then we have the fan and the fan blades.

Then this is a gear box mechanism which is here, and we have the fan stack door for people to move out and inspections and so on. Then this is a motor, this is a pedestal for the motor, and this completely you have a fan deck slab for people to move around. So, here if you see there are all people coming in, you need electrical wires and cables and so on, in order to move the fan, operate the fan motor and gearboxes.

Civil work is already there, these instruments are primarily the mechanical equipments. And we also have so many instruments to check the noise on the fan, to check the vibration of the gearboxes and so on. So, there are other instrumentation assemblies are also there ok.

(Refer Slide Time: 09:58)



Level	Team/Group	Components	Activity list	ID
Drive Assembly	Mechanical	Fan	Fan design	DA
		Motor	Motor design	DB
		Gear box	Gear box design	DC
		Drive Shaft	Drive shaft design	DD
		Accessories	Details of Equipment Handling System	DE
			Details of removable walkway, handrail & trolley	DF
			Details of ladders, doors & hatches	DG
	Civil	Accessories	Details of insert plates, anchor bolts	DH
		Fan deck slab	Design & details of slabs, grid beams & pedestals at fan deck level	DI
		Fan stack	Design of fan stack and ring beams	DJ
	Instrumentation	Level Switch	Data Sheet of Level Switch	DK
		Accessories	Instrument Location & Cable Routing	DL
		Vibration Monitoring System	Data Sheet of Vibration Monitoring System	DM
	Electrical	Accessories	Cable trays & trench layout	DN
		Illumination system	Illumination system (lighting) of CT	DO
		Earthing & lightning	Earthing details & lightning protection	DP

So, now what do I do, as I told you first is you have to get the list of elements ok. The project is defined, I am going to work only on drive assembly level that is very clear; then the end is decomposition level, decomposition I am going to use the WBS structure only.

So, first is level that is how the project they are very comfortable in working. So, I am also going to show in the same way. So, drive assembly level that is a level with which it gets decomposed; then you have mechanical team, civil team, instrumentation team, and electrical team, there are four teams in the particular project. And you know what are the components here fan, motor, gearbox, driveshaft, accessories in the mechanical side.

For civil discipline, there are accessories, fan deck slab, fan stack and so on; In Instrumentation, level switch, accessories, vibration monitoring systems; electrical again accessories, elimination systems, and the earthing and lightning system ok, so these are all the components. So, I have one-one activity identified for all the components, never it is one-one; but in this case it happened to be one-one. So, I have fan design, motor design, gearbox design, drive shaft design, details of equipment handling system, details of removable walkway, handrails and trolley, details of ladders, doors and hatches. Primarily the layout and all these has to be shown. So, these are all the list of activities I have for the mechanical discipline ok, in the drive assembly alone.

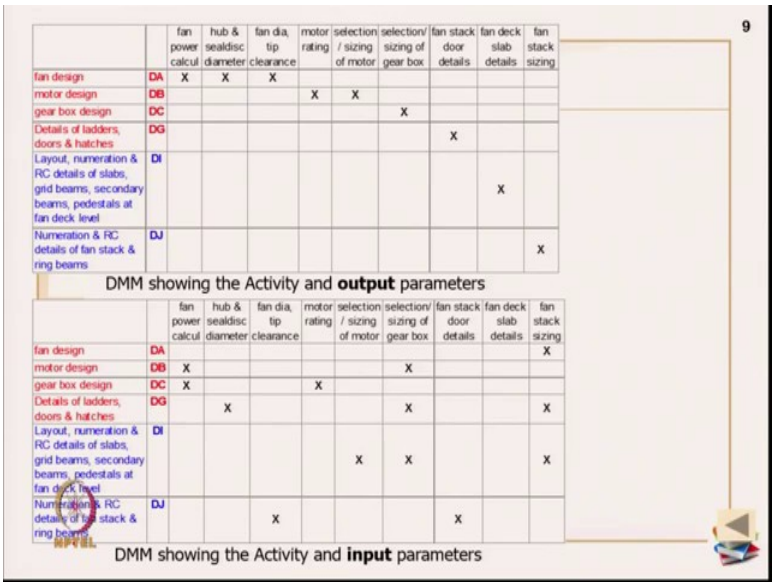
In civil work details of insert plates, anchor, bolts for all the cable trays and so on; then design and details of slabs, grid beams, pedestals at fan deck level; then design of fan stack and ring beam. So, these are all the activities in the civil discipline.

Then for instrumentation data sheet of levels switch, instrument location and cable routes, data sheet for vibration monitoring system, so these are all the activities we have in the instrumentation. For electrical, cable trays and trench layout, illumination systems or lighting system of the cooling tower, earthing details and lightning protection; so accordingly we have almost like 16 activities in the entire drive assembly level.

Since it is very big an activity list, I have given an activity ID for the same starting from D; because it is a drive assembly a level, so I have used D as a code. So, D stands for drive assembly and A B C D are all the list of activities ok. So, I have used simple WBS for the decomposition. The next is how do I, so with this I can easily get my skeleton DSM and I can fill in my relationships.

But what happens is with the way you wrongly identify a relationship, or you put the relationship in a different place or something or you miss out identifying a relationship; you get a different result altogether. In order to avoid that, we are using a new methodology. So, this methodology is nothing, but multiplying 2 matrices. So, that I get a DSM ok.

(Refer Slide Time: 13:14)



		fan power calcul	hub & sealdisc diameter	fan dia, tip clearance	motor rating	selection / sizing of motor	selection/ sizing of gear box	fan stack door details	fan deck slab details	fan stack sizing
fan design	DA	X	X	X						
motor design	DB				X	X				
gear box design	DC						X			
Details of ladders, doors & hatches	DG							X		
Layout, numeration & RC details of slabs, grid beams, secondary beams, pedestals at fan deck level	DI								X	
Numeration & RC details of fan stack & ring beams	DJ									X

DMM showing the Activity and **output** parameters

		fan power calcul	hub & sealdisc diameter	fan dia, tip clearance	motor rating	selection / sizing of motor	selection/ sizing of gear box	fan stack door details	fan deck slab details	fan stack sizing
fan design	DA									X
motor design	DB	X					X			
gear box design	DC	X			X					
Details of ladders, doors & hatches	DG		X				X			X
Layout, numeration & RC details of slabs, grid beams, secondary beams, pedestals at fan deck level	DI					X	X			X
Numeration & RC details of fan stack & ring beams	DJ			X				X		

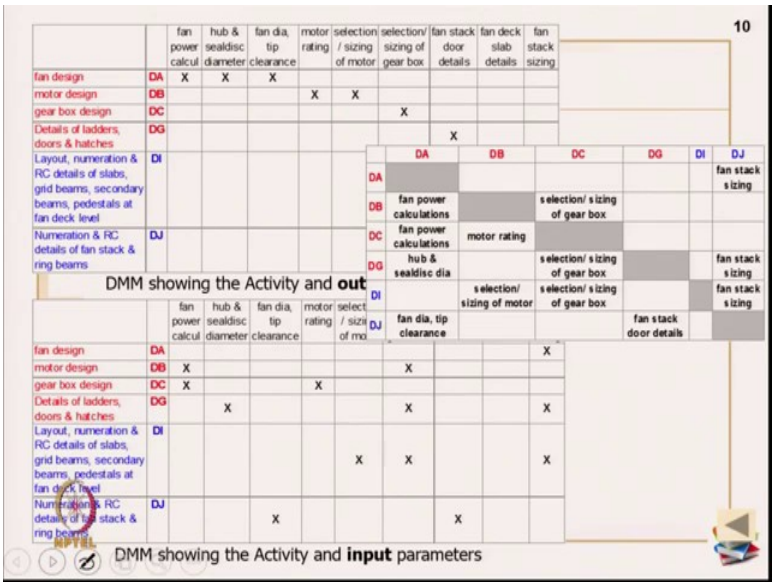
DMM showing the Activity and **input** parameters

So, these 2 matrices are variable DSMs or we call it as a DMM, DMM stands for Domain Mapping Matrix ok.

So, the DMM is primarily the domain mapping matrix which stands for which actually has a rectangular shape and it is not a square, wherein it maps relationships between two entities. So, one can be as here, it is primarily a list of activities on the rows and along the columns I am having a list of parameters ok. So, with this you can form all the relationships and what happens here, any activity will have a set of outputs at the end of the activity execution and activities also need inputs from other activities ok.

So, every activity needs a set of or list of inputs. So, when you multiply these two you will actually get an activity DSM, that is what is the main purpose behind. And logical way of doing this also I will show you right now, manually also you can do, you can make this as binary matrix and multiply the 2 matrices you will obviously, get another matrix ok. So, for example, let me read through. So, fan design, so this is for output parameters and this is for input parameters. Fan design can release fan power calculations which is required by motor design and gearbox design; and fan design can release hub and seal-disc diameter which is required for design of ladders, doors and hatches. So, fan design also releases as fan diameter and tip clearance which is required for numeration and RC details of fan stack and ring beams. So, like this you have to go through this.

(Refer Slide Time: 14:52)



So, what happened is? So, I am having a DSM like this. So, these are all my activities, I wanted to form an activity DSM. So, DA, DB, DC, and so on; as I told you for beginners it is better that you write what is the X really representing for, till you go for validation and check for the whole flow. Once the whole flow is really defined, then you can cut off all the values and you can replace it to X marks, which is very easy to do.

For example, so DA is releasing fan power calculations, which is required for motor design and gearbox design. So, what I have written here is DA to DB, what is the X actually there, it is a fan power calculations, DA to DC what is a X really representing it is a fan power calculations. Now DA also releases hub and sealdisc diameter which is required for DG, which is details of ladders, doors and hatchers. So, same way DA to DG hub and sealdisc diameter is the parameter which is traversing between from DA to DG. Now same way the fan design also releases fan diameter, tip clearance which is required for activity DJ which is a numeration and RC details of fan stack and ring beams.

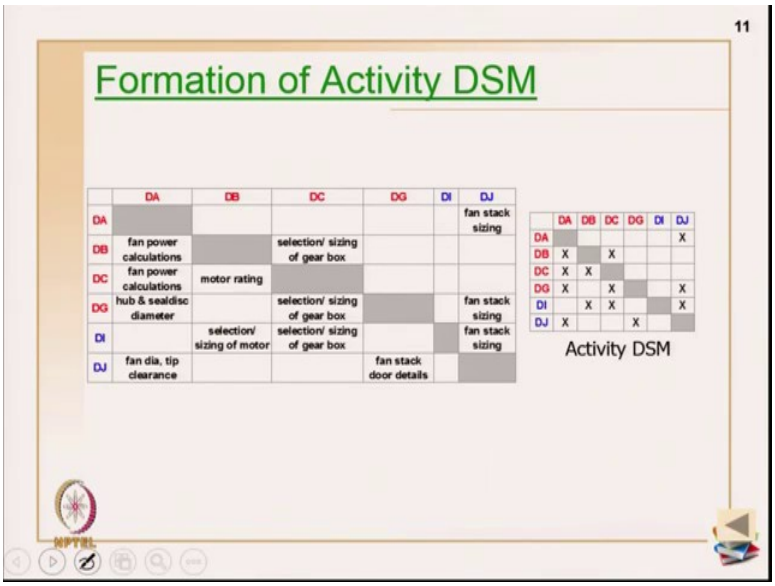
So, what happens is DA to DG, primarily the fan diameter and tip clearance is what is a input which is travelling from the one activity to other activity. So, same way, we can travel through the entire activity DSM; motor design releases motor rating which is required for gearbox design which is shown here. Now this motor design also releases

selection and sizing of motor which is required for activity DI, which is the layout numeration and details, so here you have written here, selection and sizing of motor.

The gearbox design releases an output called selection and sizing of gearbox. The selection and sizing of gearbox is actually required for motor design, it is required for details of ladders, doors and hatches; it is also required for layout numeration. So, it is required for three activities. So, selection and sizing of gearbox is required for DB DG and DI.

Now the details of ladders, doors and hatches releases information on fan stack door details, which is actually required for activity DJ which is the numeration and RC details of fan stack and ring beams ok. So, which is written here and accordingly the entire matrix is filled. So, this is how the activity DSM is filled.

(Refer Slide Time: 17:35)



So, now I have got the same matrix, I have shown it here. Now once you take it to validation, it is easy for somebody to go through ok, fan power calculations this is required for here. So, it is easy for somebody to validate this matrix and once a matrix is validated then you replace it to X marks for your further calculations ok. So, now, formation of activity DSM is really done ok.

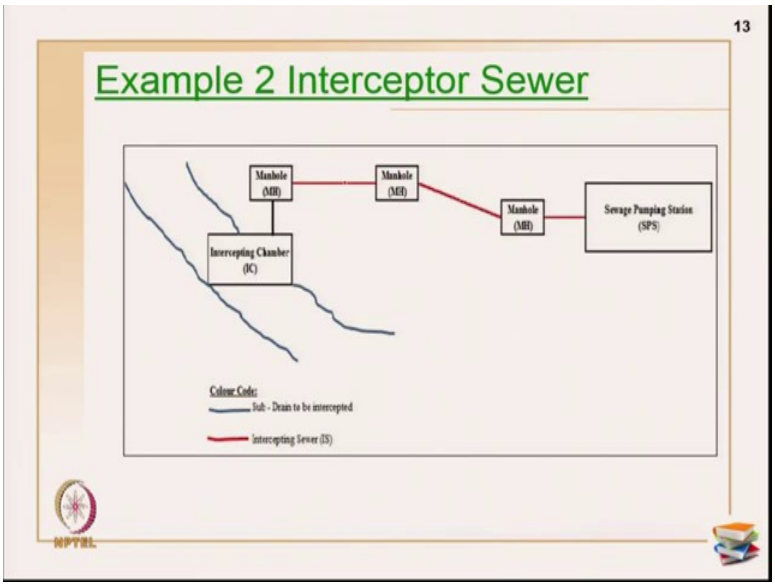
So, so far we started up with defining the scope of the project, then decomposing the project that I have shown you the levels ok, teams, components, then activity list; and

then use variable DSM, so the other name I can say is DMM which is domain mapping matrix. And I am having a rectangular matrix 2, rectangular matrix come; then primarily you have to multiply if you want to mechanically do or you can go through the logic also I have explained you that, you do that. And then you are getting this matrix convert into X marks, so that you will get the activity DSM ok. So, which is what I have shown.

Now in this, define the scope, define the decomposer system. So in this decomposer system I can stop at various levels. In the previous example I have stopped at activity levels, now in the next example I will show you how to form a team DSM or a component DSM using the same methodology.

Now whatever you have stopped you wanted to form the type of DSM, you can decompose elements in the same way, list the elements in the same order and you can still collect data on relationship and you can still go ahead with the matrix formation.

(Refer Slide Time: 19:20)

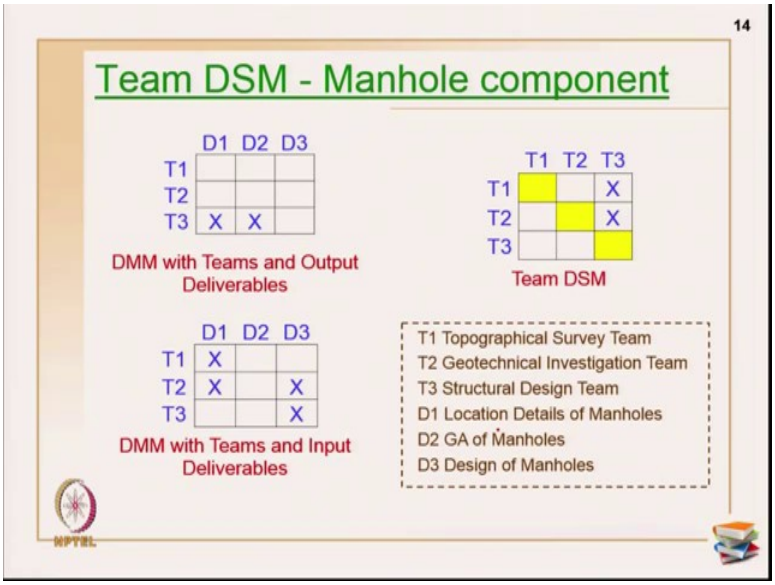


Now, I am going to take the next example; the next example is on interceptor sewer ok. This is a very simple example; I have shown this example in different form in the earlier classes.

Primarily I have four components ok, I am going to work at component level. So, I am only showing the components, I have intercepting chamber, manholes and sewage

pumping stations, lifting stations are also there, but I am not showing right now in this particular work, and this red line is primarily the intercepting sewer ok.

(Refer Slide Time: 19:46)



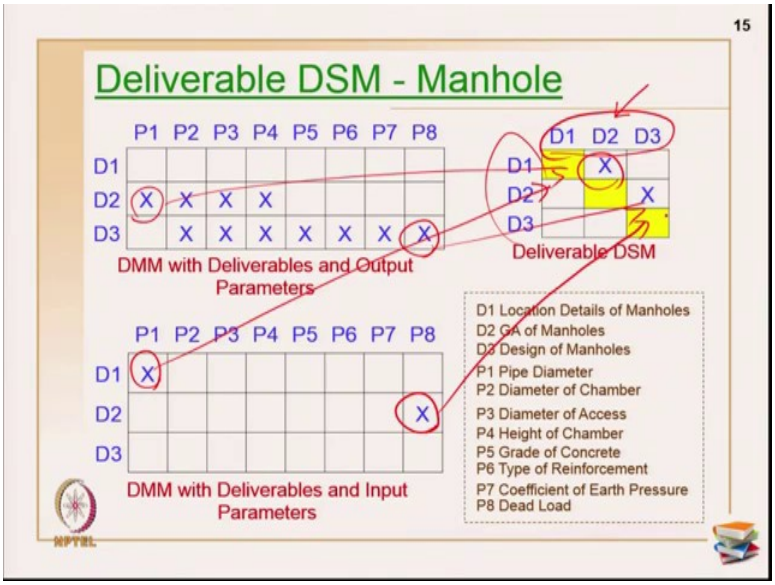
Now, for the particular Manhole component alone I am going to show the team DSM formation ok. So, now, these are the 3 teams T1 T2 and T3 and these are primarily the drawings D1 and D2 and D3 are the 3 drawings ok. If you see here, the identification is given in the legend here, topological survey team is T1, geotechnical investigation team is T2, structural design team is a T3 and I have details on deliverables.

So, D1 is location details of manholes, D2 is GA of manholes, GA stands for General Arrangement and D3 is design of manholes. So, this is a first DMM which I have formed with output deliverables. So, T3 is actually giving two deliverables which is, D1 and D2 ok. Now these deliverables are required, there are several deliverables required for the teams ok. Let us see on the input deliverables, so D1 is required for T1 and T2, D3 is acquired for T2 and T3 ok. So, now, if you see here, I am going to form a team DSM; now I have not used all my X marks here, I will tell you that also later. So, T3 is releasing an input called releasing an output called D1 which is required by T1 and T2; T3 is actually having a dependency relationship on information with T1 and T2 ok. So, this DSM is done ok.

So, I have used only the few relationships here, other relationships I have not used. So, which is what I said, when you actually avoid this X mark and then write properly what

is it; like I showed you in the last example, it is easy for you to map the complete or go through the complete flow of the process, design process and map all relationships ok. And the other problem that happened here is, I have used a small example so many of the relationship is got missed out, when you are really building up a complete a design example all relationships can be easily be identified and shown ok. But the purpose of this example is to only show the, how to form a component sorry team DSM for the manhole component ok; same manhole component how to form a deliverable DSM ok.

(Refer Slide Time: 21:59)



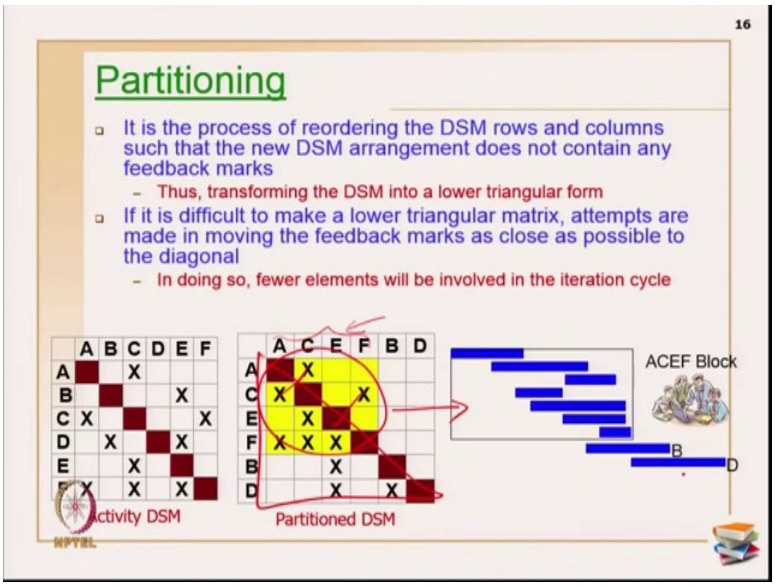
So, I am going to use DMM again, but there is a list of deliverables and output parameters.

So, now, these are the deliverables same the deliverables I have location details of manholes, general arrangement of manholes, design of manholes; and I have identified 8 parameters for this all these 3 deliverables. Pipe diameters P1, P2 is diameter of chamber, P3 is diameter of axis, P4 is height of the chamber, P5 is grade of concrete, P6 is type of reinforcement, P7 is coefficient of earth pressure, and P8 is dead load. So, I have all these parameters, same logic still works good. So, I have list of deliverables, I have list of parameters here. So, these are all my X marks identified for the parameters, same way for the input parameters.

So, now, what happens here is, I am going to use for my D2 is actually releasing P1 which is required for D1 which is identified here. So, D2 is releasing P1 which is

required for D1 which is identified here ok. So, now, D3 is releasing P8 which is required for D2 which is identified here ok. Other parameters did not come within the scope, but the idea is to convey to you. So, the logic for formation, if you want to identify a complete DSM is, go through decomposition, WBS will still hold good for you and then formation of DSM in, so sorry. So with the help of WBS you can identify all these levels ok. It can be level 1, level 2 whatever it is, drive assembly level or you can go through team level or component level, activity level. I have shown you several cases right now, so you can stop at any level. WBS will help you to identify only this, once this is identified for identifying X marks you can multiply the variable matrices and you will still get the relationships identified. Then go through with the experts, see whether everything has been identified and then you have to validate your matrix formation ok; because if you identify a wrong matrix then obviously, your results are all going to fail very bad ok. So, that you have to take care.

(Refer Slide Time: 24:32)



So, that is all on about matrix formation. So, now, the next step is Partitioning ok. So, as I told you first step or first operation in your matrix operation is formation of activity DSM.

Once the matrix has been formed, the next step is primarily the partitioning processes. So, what is partitioning, I think I gave a quick info earlier also while I introduced in the lecture. So, partitioning helps you to identify what is the relationship of the entire project

and wherein there are no interdependencies or cycles or let me put it another way. So, when the project has independent and dependent activities those are all identified in the partitioning process and wherever they are cycles, the interdependence, they are all group together as a block formation.

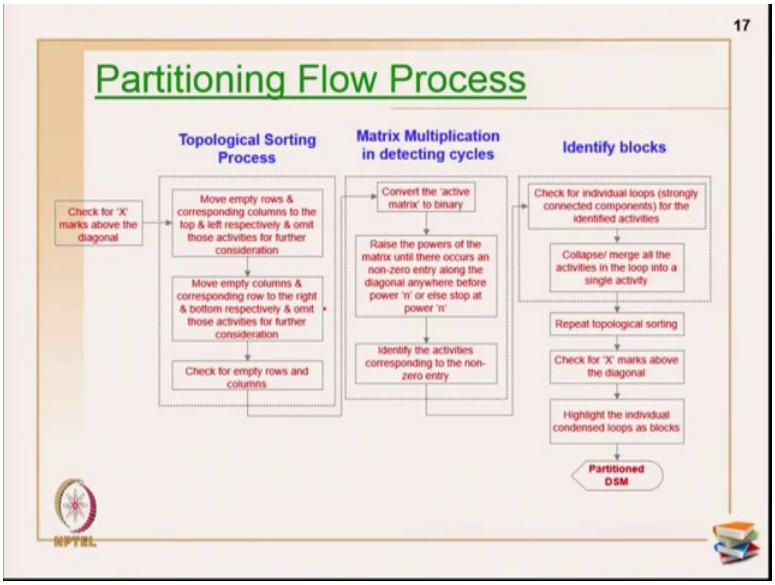
So, in a complete project, I will know what is a sequence of my independence and dependence which are all collected together and then I know where is my block formation happening ok. And the methodology also I will explain, which is a very simple method on matrix multiplication; I will show you that and along with an example, so that you can follow me along with example and various steps how the matrix gets changed ok. That is all is about the partitioning process; partitioning never talks about relationship within the block and also it never talks about sequence within the block ok. This A C E F, so I have shown here, partitioning will not talk about this sequence; so you have to go through the next step in order to identifying what is the sequence there ok.

So, let us now you see the definition of partitioning, it is a process of reordering the DSM rows and columns, it can be component, it can be teams, it can be activities, it can be deliverables; whatever you have identified you are only going to reorder in such a way that the new DSM arrangement does not contain any feedback marks or primarily you have to minimize it into lower triangle ok. So, thus transforming the DSM into your lower triangular form ok; so primarily I will be getting a lower triangular matrix ok, if I have only an independent and dependent relationships. Now what happens is, if it is difficult to make a lower triangular matrix because of the interdependencies and cycles, attempts are primarily made in keeping those elements identified as a block formation ok. So, still there are methods which can make you to keep these X marks close to the diagonal, but partitioning never attempts on that ok.

In doing so only few elements will be involved in the cycle, so try to minimize this formation in self identification of block itself. Now what is the meaning of this block formation, that also you should know. So, this implies these four; suppose if these are all activities, these four activities have to be work together till the completion, so that because each of them reads input from each one of the other ok. For example, ACEF is primarily a block ok, what is the meaning of the block? The teams have to work closely together like this in order to finish the completion of the particular scope of the work ok;

and then activity B and D. So, that is a meaning of the, logical meaning of the partitioning block formation ok.

(Refer Slide Time: 27:56)



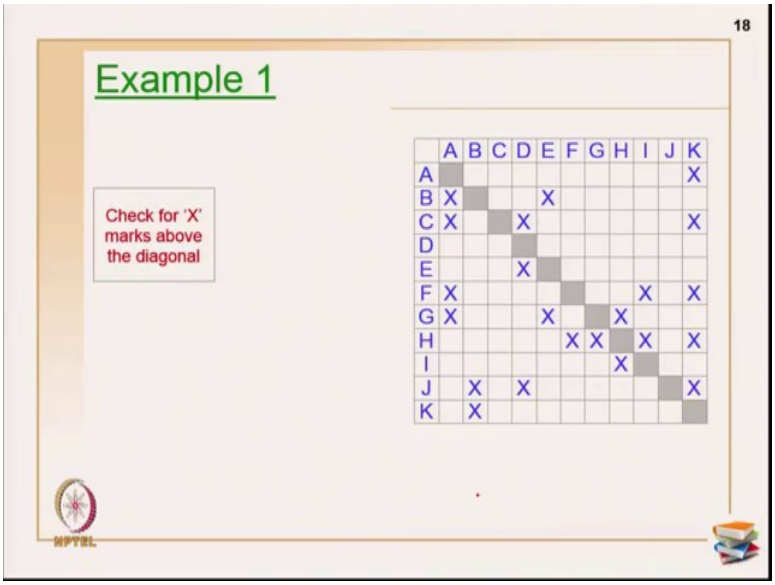
Now, what is the partitioning flow process? Now there are several methods for partitioning as given in the websites ok. If you go through the DSM website as such, you will be there are so many methods like path searching, matrix multiplication, or we have triangularization matrix like this n number of methods available ok.

So, for identifying the sequential the parallel and sequential activities, topological sorting is the best way to do. And once you identify the topological sorting and identifying that there is a big block, but I do not know what are all there within the block, then you have to choose one of the methods like path searching or matrix multiplication and so on ok. But it is not a straight forward method, you have to work in cycles; primarily there is an iteration inside, which with an example. I have taken an example in such a way that, it can cover all the problems which you have in the examples ok. So, within this cyclic process also I will explain properly.

Now partitioning flow process, there are three broad items here; one is topological sorting, primarily to identify all my sequential and parallel activities. Then I do matrix multiplication, matrix multiplication or take off this concept and put path searching can help you to identify or trace the cycles only; once you have trace the cycle, then how to group them into blocks again repeat the process on topological sorting and matrix

multiplication until you identify all the cycles and all the activities ok. And at last you get the partitioning DSM; this is the overall flow on the partitioning process ok.

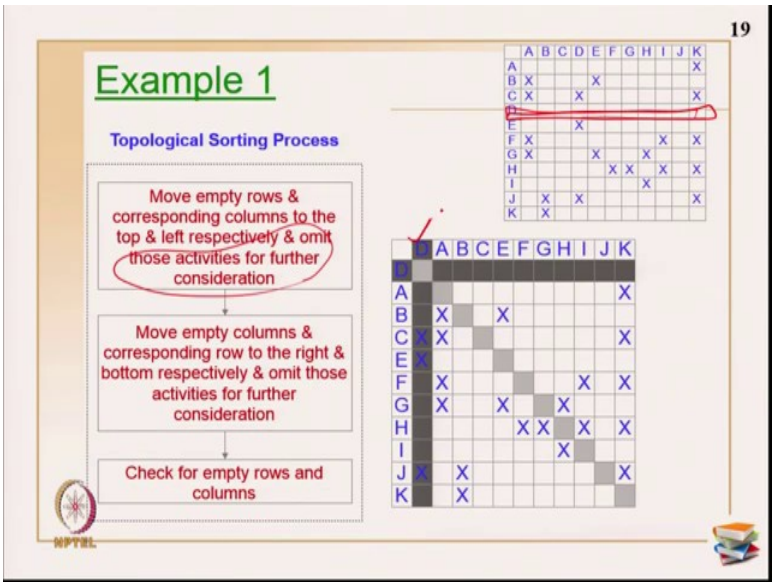
(Refer Slide Time: 29:41)



Now, let us go through with the hypothetical example. So, for explaining this partitioning process, I have taken a hypothetical example ok.

This is a generic A B C D example, I have taken A till K as a simple example; because I have to show all these n number of steps, I have taken a little bit bigger example only. What is the first step as shown earlier, check for X marks above the diagonal? Do I have X marks above the diagonal, yes I have lot of X marks above the diagonal, which implies this is not the right sequence for you to execute the whole block. Suppose if there are no X marks above the diagonal. So, what it means is, the matrix is already partitioned ok, already you know what is the best sequence to operate the entire flow ok; that will be very sure if you do not have any X marks about the diagonal. Then directly you can go to the last step in saying the matrix has been already partitioned ok.

(Refer Slide Time: 30:43)



Now, since I have X marks, so same example I have shown you in a smaller form in the top, so that you will follow the slides.

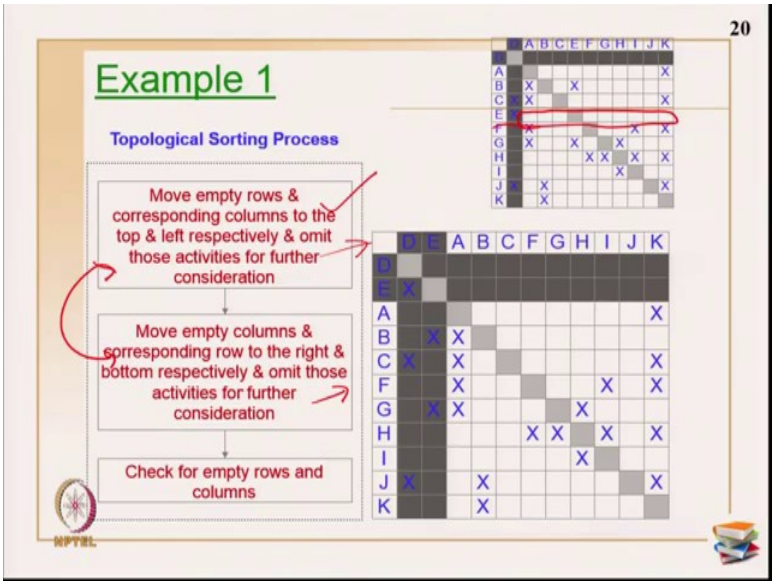
Now as I told you, once we have X marks above the diagonal; the first step is to do topological sorting ok. Topological sorting is the graph theory terminology or concept. So, what does it say, I have written very simpler steps for you. So, primarily you have to identify all empty rows and columns or rows and columns in order identify the start and the last activities ok. So, when I am having a starting activity; obviously, I would not have any X mark along the rows. So, that will be shown as an empty row and when I am at the last activity, then I am not going to give any inputs to other activities. So, I may have empty columns and, but I may have lot of X marks along the rows ok. That is the meaning of topological in order to identify the starting parallel activities and the last activities. Primarily the first activity, last activity, I am going to trace out now with this method ok.

So, first step is move all the empty rows and the corresponding columns to the top and left respectively; and omit those activities for further consideration ok. Now, let us go through this example, where are the empty rows right now; activity D has an empty row. So, there is a complete empty row here ok, there is nothing given in the activity D. So, what I am going to do is? So, I have identified that activity D has an empty row, you can use excel sheets for doing or there are software also available which can help you to do

the partitioning process ok. So, I have moved my activity D to the primarily to the top and left, because I have to pull out both my rows and columns accordingly outside. So, just cut off that row and the particular column and move to the top and push all the other activities. Then you will obviously get this matrix. If you want to do manually; otherwise you can create you can run a small VBA program and you can still go through the exercise ok.

So, now, what we have to do is, omit those activities for further consideration. So, in the first time I have written, move them to the top and left and then do not include them in the further for the calculations ok. I have identified D as my first activity, and I am going to ignore this in my further analysis ok. Now what you have to do? You can repeat the process of identifying empty rows again or you can go to the other step as identifying empty columns and you can swap this up and down ok. In this particular example, what I am going to do is. I am going to identify all the empty rows after omitting the activities; and at the end I am going to identify my empty columns and this I am going to repeat it again and again ok.

(Refer Slide Time: 33:38)

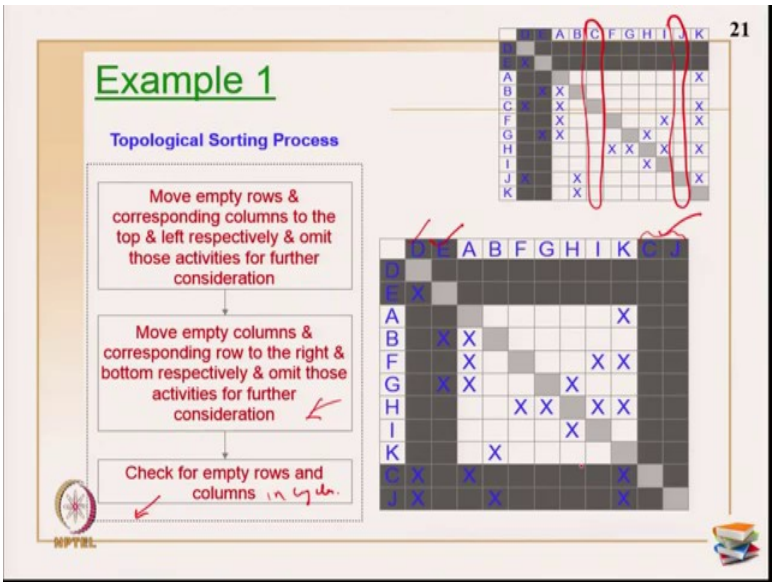


So, now let us see the next step. So, what I am going to do is? I am going to identify all other empty rows and rows also. Now if you see, after omitting this activity where are my empty rows, this activity E has completely no X marks along the rows. So, activity E is primarily my second activity in the list ok. So, D is giving inputs to activity E, but it is

not the first activity, but it is a second activity ok. So, push the activity E next to activity D and then omit this from further consideration ok.

Now do I have any other empty rows? So, in this example I do not have any other empty rows. So, primarily I have finished my first step on: move empty rows completely; which primarily is the starting activity or next sequential activities ok. I have identified all the first few start activities ok; after then the block starts, and you are not aware of where the block is and so on ok. Now I am going to move this. So, I am going to finish this step in cycles and then I am going to do this. But as I again I am telling you if you want you can swap these two in order; you do this first, do this second, then you can do this. Primarily what I have done is, I have finished all, this step is over now, I am going to go to the next step now.

(Refer Slide Time: 35:01)

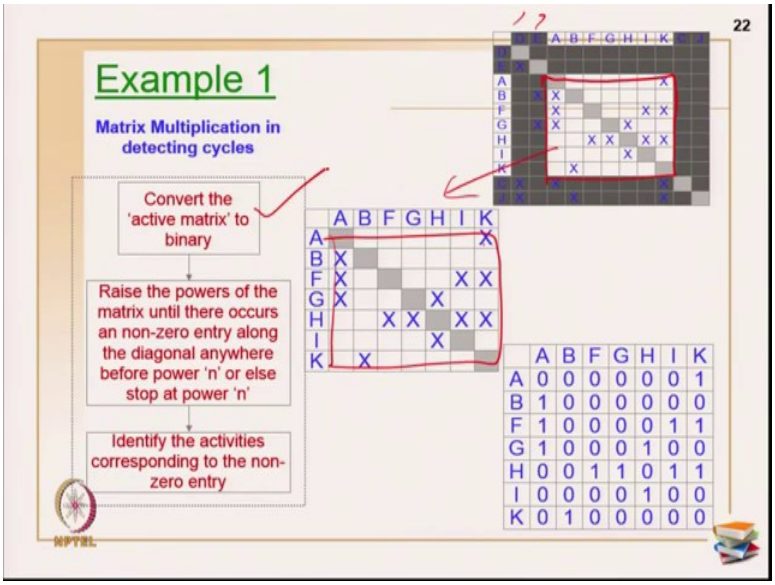


This is my last step I have shown this picture in the top corner. Now move all empty columns and corresponding row to the right and bottom respectively and omit those activities from further consideration ok. So, what are the activities that are empty; I have activity C which is empty here, I have activity J also which is completely empty ok. This looks like C and J are parallel activities; but they are in the last in the sequence ok. So, I can move C and J completely to the last. Now what is the order of C and J, that is there is no restriction here; because both are empty, both are parallel activities, there is no worry

if you want to place C first and J first or you want to place J first and then C first, there is no problem at all ok.

So, now, I have moved C and J to the last. So, now, what happens is, I have identified D as my first activity, second is E activity, I have also identified two parallel activities in the last as C and J. Now again I have to repeat this process to check any other empty columns are there. So, any other empty columns are there, it looks like there are no other empty columns ok. So, this process I have to repeat in cycles, which is what I said ok. This has to go in cycles, until completion. Now within this then have got a bigger block and all empty rows columns are identified ok. Only the start last activities are identified. Now you have to find out what is there within this particular stretch ok.

(Refer Slide Time: 36:39)

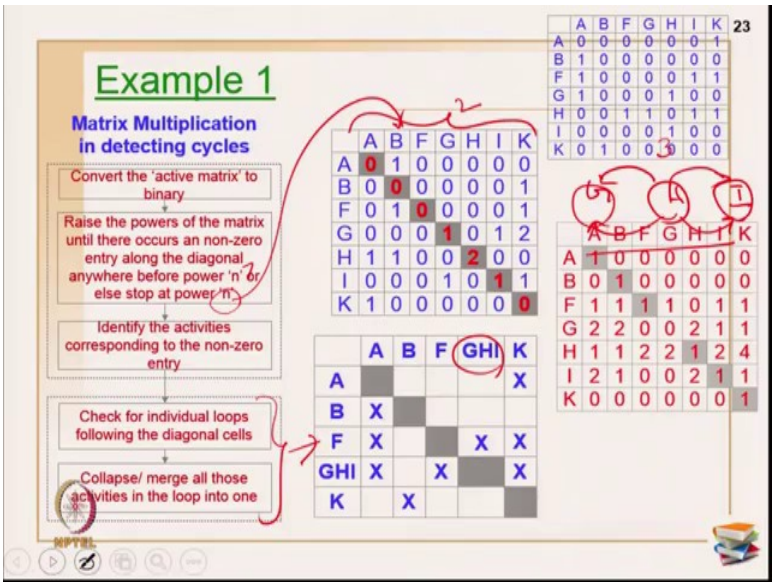


Now, what is the next step? So, topological sorting process is over ok. Next step is, matrix multiplication in identifying cycles. So, now, what you have to do? I have got this matrix, now I am going to give a new name for this matrix. So, this is primarily the active matrix, which implies the matrix for consideration is only this portion for you, because you know D is first activity, E is second activity, C and J are the last activities that is known to you very well.

Now your main botheration is to know what is a sequence within this, ok. Now what you have to do is, convert the active matrix into a binary matrix. Now this is my active matrix, now how do I convert into binary; while explaining the types of DSM I have told

you this. So, replace all these X marks as 1, if you are working on excel again it is very easy, select for all these this particular cell ok, search for X and replace with 1 you will get all ones; empty cells all if you put 0, you will easily get a 0. So, I have just converted all these into a binary matrix. Now I have my binary matrix. So, this step is really over ok. So, take out the active matrix and I have converted into binary and I have the binary matrix right now, ok.

(Refer Slide Time: 37:57)



Now, what do I do? Raise the powers of the matrix.

So, this is my example. Raise the powers of the matrix, until there occurs a non-zero entry along the diagonal anywhere before power n or else stop at power n. Power n, n primarily I am using an alphabet which talks about the number of activities here ok. For example, in this case I have 7 activities, so I have to stop till power n, whatever happens; or well ahead of power n, when I am able to get some non-zero along the diagonal ok. So, I have raised, I have a binary matrix which I have shown you in the previous case. So, I have done primarily one multiplication only, which we call it as power 2. Power 2 is the binary square ok. So, I have done multiplying only once.

And what happens is, I identified a non-zero entry here ok. So, this implies G is here, H is here and I here. If you are not very comfortable in identifying or tracing the loop, you have to do the tracing by go throw the relationship and then identifying where is the real loops ok, because I have got values as 2 also. This 2 implies this H is on two cycles and

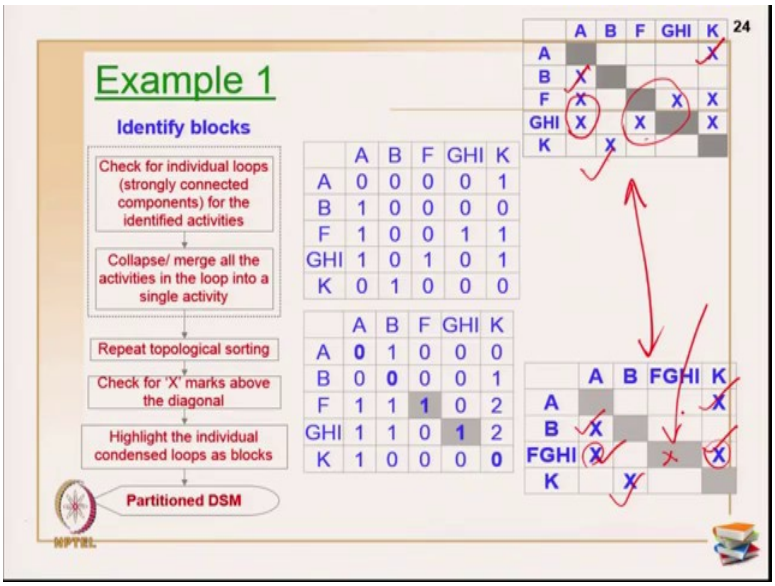
G implies it is on one cycle and I implies it is available on one cycle. So, how does this relationship look, you can go through later after cross checking with the example. So, this is what is the tracing on the whole loop. So, this is primarily we call it as overlapping cycles. So, H is primarily is in the midway between G and I, in the length of cycle of length 2 ok; that is what we have identified in the matrix square or adjacency matrix raise to powers 2 ok.

So, now, what happens? Once you have identified the activities corresponding to the non-zero entry, I identified it G H I are in those activities. Check for the individual loops that also I have done, sometimes I may get values here there, this maybe in some other block; this need not always be together in one block only it can be two three blocks of cycle of length 2 can also happen. So, you have to check for individual loops; the one way to do, again I am repeating go through with the original example and trace for the cycle, so that you will get the cycle ok. Then what you should do? Collapse or merge those activities on the loop into one. So, what I have done G H I since it is all in one group only, I have merged the G H I into one block. So, I have made it as one and I have merged the relationships also into one ok. So, I have now the new matrix which is like this ok. Now what you have to do, again you may have.

So, let me also tell you what is the problem? If I have not stopped at this step, ok. You should also now, you may have a question why should I go through this merging and this process again and again; why cannot I go till power n directly and then identify all the bigger cycles or loops. You would not be able to identify; for example, I will tell you this is raised to power 2 ok, the same example I am raising it to power 3, I am getting A B F as like this and then H I K as something.

So, the minute you go on without merging the first formed loop ok. So, the first time you are getting a non-zero entry you have to stop there; and then you have to identify the cycle and then proceed after merging only. If you do not do, there is no use in the matrix because these loops will give new values and then you would not be able to identify properly the loops, ok. So, that is what I want it to show you on what happens when you raise it on power 3, you can cross check with any example later ok.

(Refer Slide Time: 42:01)



So, now let us move on. So, what you have to do after this. So, once you identified the loops, you got a smaller block within the bigger group; and what you have to do is, you have to again do the topological sorting ok. So, which is what I have written here, so merge them into single activity; then repeat the topological sorting, then check for the X marks above the diagonal; topological sorting generally talks about that only.

Identify the highlight the condensed loops as blocks, if there is nothing else available ok. Now this also goes on in cycle. Let me show you with example then you will understand better. The same example as shown in the previous slide I have taken it up here; now what you have to do, there are X marks above the diagonal. And what about topological sorting and I do topological sorting, I do not have empty rows or columns ok; because you merged a block completely into one, you may also get empty rows and columns now ok. There maybe small blocks within this block also, you have to identify everything ok, so that, you will identify when you are doing the topological sorting again and again.

In this case, nothing happened because there is no empty rows or columns are seen in the example; then what you have to do, you have to identify the remaining cycles or loops ok. So, I am converting into binary, same process ok; then when I raised it to power 2 identified that, there is a block of cycle length 2. So, F is dependent on G H I, and G H I is dependent on F which is what I know ok. This if you are very an experienced and not a beginner, you will also identify this with the help of this X marks also. If you see here

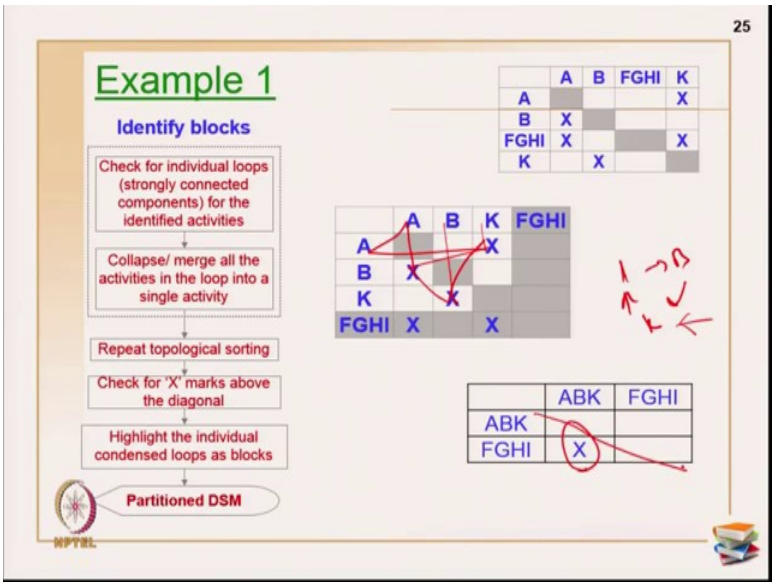
with X mark here it shows that these two activities are interdependent ok; you need not even do the multiplication, without that also I will understand; but if you are new to the whole process better you do all the steps.

So, this I have identified, because I have got 1, 1 of this implies F G H I or in cycle. Now what am I going to do, I am going merge the F G H I as one. You will see here the merging, primarily I wanted to show both the merging together. So, if you see here since I merged the F G H I in to one, you will see the other relationships are not missed, but this relationship within relationships within the F G H I are all got done in the sense they all are not visible ok; that is the meaning of showing the merging.

So, K is, so K gives information to A that is seen here, A gives information to B that is also seen, A gives information to F or F G H I that is also seen here ok. Now F gives information to G H I and G is giving information to F G H I you need not show, because there is no relationships which is done.

This you need not even worry ok; when you are merging it, just merge both the X marks together then obviously, you will get all the X marks identify ok. Now K is giving information to K to F and to G H I which I have shown it up here ok, now B is giving information to K that is also coming here, which is also seen here ok. So, all relationships you have to identify, the relationships within the particular activities in the merging you need not show. That is the meaning of what I am saying as merging into one; because you know that there is a block, so you wanted to keep it as one activity and then start repeating the process ok.

(Refer Slide Time: 45:42)

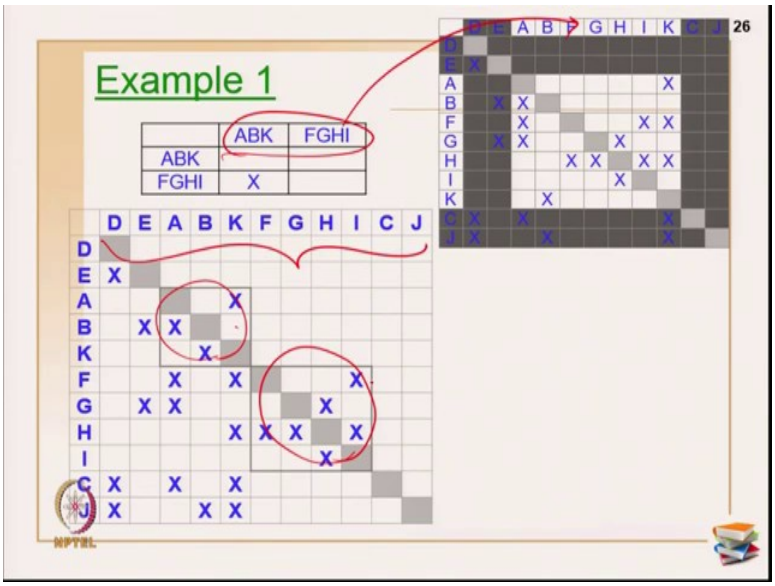


So, now I am going to repeat the process. So, I have four activities right now A B F G H I block and then activity K. Now when I repeat the topological sorting what happens, see here the F G H I block is primarily in the last and A B K has come to the front ok. Now there is a cycle within this A B K which you have to identify; because in the further topological sorting what happens, there is no empty row or column so obviously, you know that there is a block in the A B K ok. Again, if you are really experienced with the DSM you will understand.

So, there is a cycle of length 3 here, because I have this pattern here. K gives information to B, A gives information to B, B gives information to K. So, primarily A to B to K which when you are raising the matrix into power 3, we will get one along the diagonal you can cross check that later. So, you will identify it as power 3 and when you are raised to power 3 and there is a cycle of length 3. So, now what you have to do, merge this A B K as one activity, F G H I as one activity; when you are merging you are getting a relationship here ok, during the merging.

So, this automatically is topological sorted ok. Otherwise you may have to repeat the process again. The automatically I am having an X mark below the diagonals, so this implies the matrix is sorted already ok.

(Refer Slide Time: 47:11)



Now, where did we start, we started upon this: the topological sorting process identified activity D as my first activity, E is my second activity, C and J are my parallel activities in the end. So, start of the project, end of the project I have identified and there are lot of activities within the middle ok. That you wanted to really explore that, for that we have done Matrix multiplication; we identified that A B K is one block and the F G H I block is dependent on A B K. If you see here are all the 7 activities in the block has been identified, yes I have really done.

So, what is a partitioning DSM I am having. So, D E is my first, A B K block is here, F G H I is here, C J is here, so this is my sequence, these are the two blocks I have got ok. So, that is primarily the steps in partitioning. It may look like you have so many steps; but I have explained you the logical meaning behind those steps also, you need not go through all the steps if you are very comfortable in seeing the relationships and working on with DSM. If you are very comfortable in plotting out the relationships on your own; then you can easily identify all the blocks, and then plot the X marks ok.

So, that is all on above the partitioning example. So, so far we have seen formation of DSM and the partitioning of DSM in the basic DSM operations ok. So, I have taken two cases; one is formation, one is partitioning. In the next class we will see the next two operations in DSM; it is primarily the tearing process, the repartitioning of the tearing processes, and iteration modelling ok.

Thanks bye.