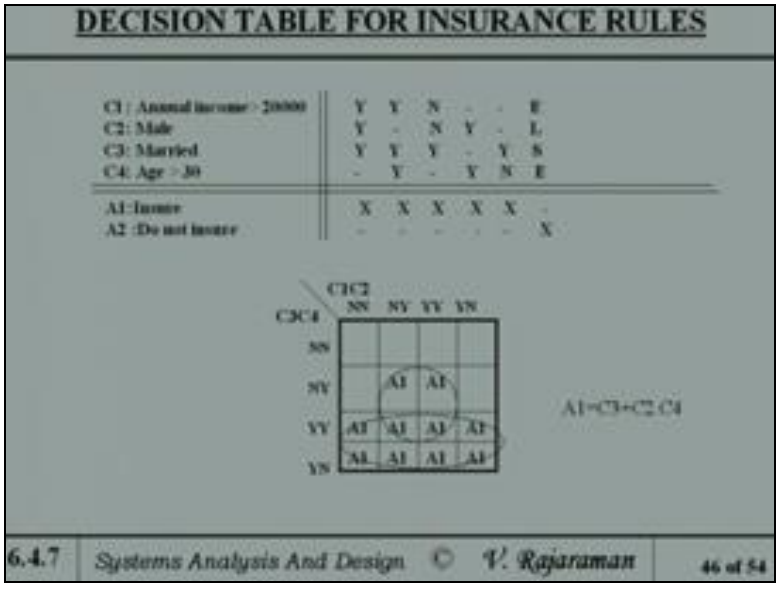
System Analysis and Design
Prof. V. Rajaraman
Department of Super Computer Education and Research
Indian Institute of Science, Bangalore

Lecture - 19

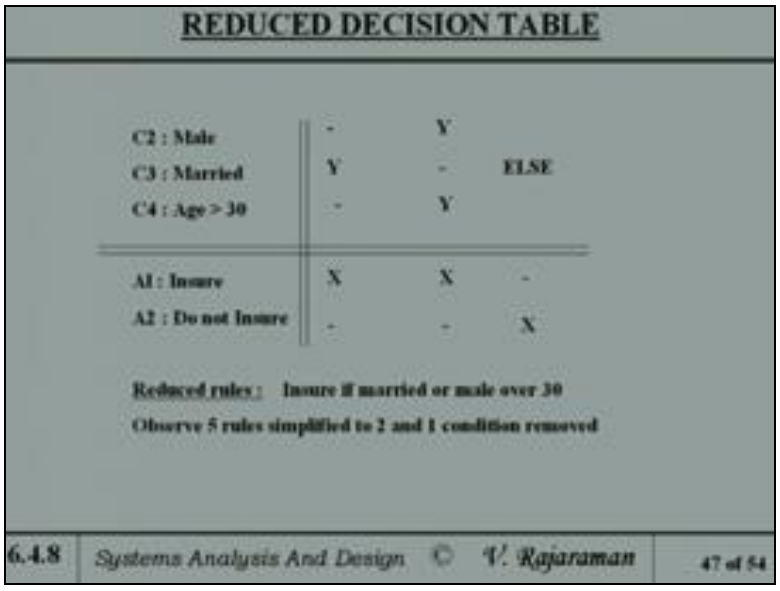
Last time I came to certain abrupt closure because I was running out of time. So, let me very quickly review the set of rules for insurance company, which we are looking at. And we started off with the set of five rules. And we found out there are four conditions. Then we mapped it onto decision table.

(Refer Slide Time: 01:42)



And then, from that the decision table we convert it to reentry old decision table and entered it into cardo map. And after having entered into cardo map, then we use the Boolean algebra simplification methods to simplify the conditions and for the action A 1. Because there are only two action in this case insure or do not insure. If A 1 is not true then it is true and vice versa. So, having done that we got the Boolean expression for A 1 as, C 3 or C 2, C 4.

(Refer Slide Time: 02:27)



Then we mapped it on to a decision table again. And this table we found as only three conditions and two rules. And so, the rules from five get reduced to insure if married or male over thirty. And you see and get solved. I mean married first rule says insure if married. And second one says if male over thirty. So, there is slightly you can make it as two rules if you want to say. One says insure if married. Two insure if male and over thirty.

Now, the point I was making at the end of the lecture last time. Is that reason why such redundant rules come in things like insurance policies. And in fact, even many government rules and other income tax rules and so on. Is that year after year they go on abounding the rules, adding new rules. And without really looking at the fact whether the new rules which are added is contradictory to the old rules or redundant and so on.

So, the organically grow without any clear logical thinking in terms of the earlier rules. Unless somebody points it out. So, when you do an analysis and particularly when you do an analysis based on Boolean algebra it turns out to be quite obvious. That the rules which are originally suggested in this particular case or that the, it as grown to such an extent that adder redundancies have come in. And that all conditions are not required.

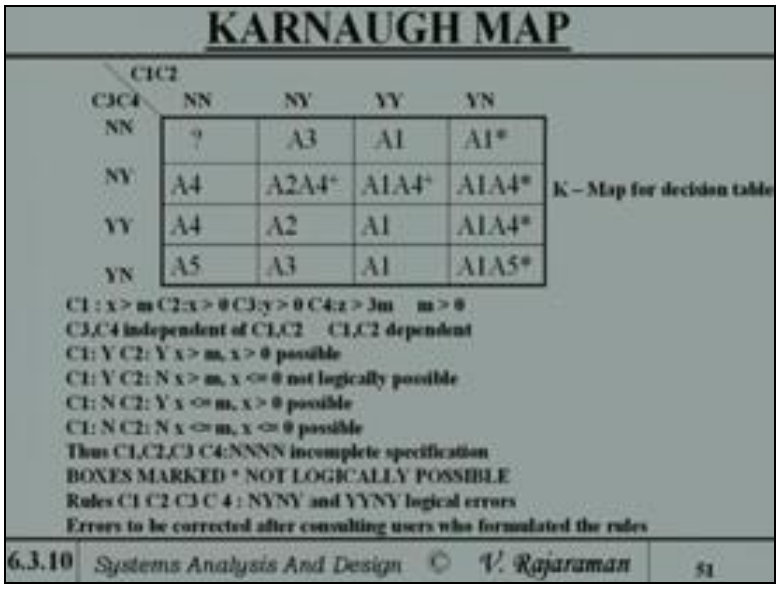
So that is the reason why it is very important. When you are given a set of Complex conditions and rules and so on, as a return word statement. You have to go through that word statement again. And apply some logical methods. To find out, if the rules are

consistent. Any contradictions are there. If there is ambiguities are or contradictions are essentially ambiguity or there is some redundancy. Extra unnecessary rules are there.

So, that is main purpose and that is what the decision table methodology allows you to do to be able to remove such incorrectness. In the for given word statement of a problem given by a customer. In fact, many companies also have similar very complex business rules. In fact, insurance rules here is a very simple example. Our insurance company giving these rules. Similarly, banks have the known rules. And so, many most average poor audio have certain business rules. Which have to be followed. Which are to be reflected by the equivalent computer program.

Ultimately the people are being removed and being replaced by machines which do routine types of jobs. Now I deliberately went to the example. Because it is very simple to a illustrate as first example. So, let me go back the original example which I started.

(Refer Slide Time: 06:14)

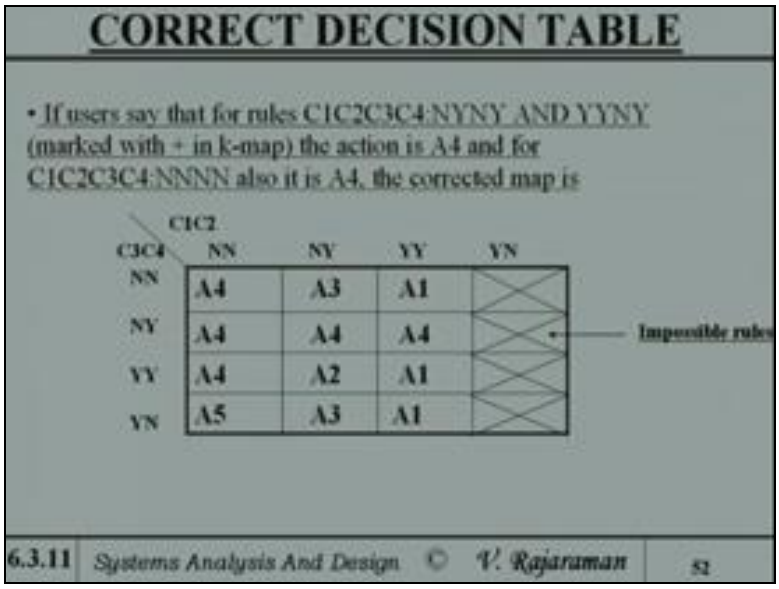


The original example which I started was the example of credit card- rules. In fact, we have this set of credit card rules put into a decision table. Then I mapped it onto a Karnaugh map. And I found that certain boxes have multiple actions. And certain box are empty.

One box is empty in this case. So, when this empty that means, the rules are that is an incompleteness. And there are multiple actions in any particular box. It means that there

are in some kind of contradiction. In these way the rules are formulated. Now, it may turn out that some of this contradictions may not occur in practice. Because of the fact that the conditions which are, which could be in this case expressed as inequalities or not consistent.

(Refer Slide Time: 07:09)



Like in this case we found that the last C one C two Y No cannot really occur here. Because of the way in which the conditions are given. So once we find out that cannot be given. That just cannot occur in practice. We cross those boxes out saying that you do not have to really take in consideration those boxes. Because no real data will ever occur which will lead to those rules. No data will ever occur. So, that means there are two advantages in this. When you take it out that means, you are eliminating those rules.

Secondly those can be used in Karnaugh map reduction. In other words they are So, called do not care conditions in the Karnaugh map. And whenever there is a do not care condition is there. That means poor audio actually occurring practice. It can be used in minimization. That is the entire column is a neighbor of some actions. So, we can combine those actions. And the previous table we found that the box M N and N which was the corner box there. And then, two other boxes N C one C two N Y C three C four N Y and Y Y N Y.

These two boxes have multiple actions. And they can really occur. In other words they are not apparent contradictions like last column. They are really real contradictions. So,

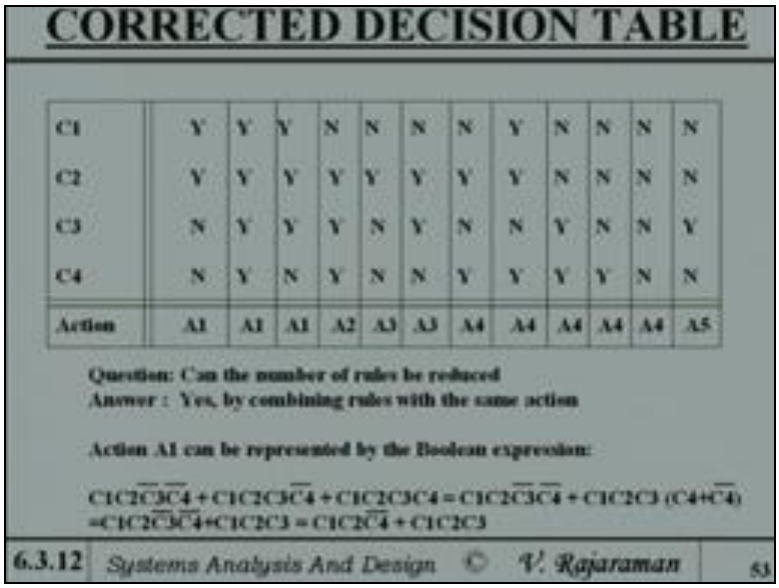
we have to go back to the manager who made up the rules. And so, that there is a contradiction. I do not know whether to take action A 1 A 2 or A 4 in one case or A 1 or A 4 or can I should I take both. Taking both is somewhat stupid because there are two different letters in this case.

The actions are send letter A send letter B and things like that. So, I got some two letters which are probably contradict to one another. So, the manager has to tell which of the actions or which, which letter as got to be sent. So, having discussed with the manager and assuming that we come up with this. That is the manager decided that the blank as got to be filled by action A 4 which is sending particular letter. That is action A 4 really is send letter D.

So, you send letter D and in fact in the other cases also send letter D. Which is letter D may be a some kind of a common type of letter. So, that whatever it is I do not know what the letter contains. So anyhow the I am assuming the manager told that A 4 is to be used try them A 1 or A 2. So, I put A 4 in both these cases then remove A 2 A 1. Now, this particular Karnaugh map has got only one and only action in each box. And it has all boxes filled up.

That means it is now does contradictions. It has no incompleteness. And now it can really is logically a correct situation. So, I map this into a onto a decision table. And end up with decision table with how many rules. Five rules because out of sixteen four are do not care. So, there are sixteen rules. But then when you look this map we see that there is a possibility of reducing the number of rules. Because of the possibility of combining actions together to eliminate some conditions.

(Refer Slide Time: 10:37)



So, in this case lets start off with the worst possible action. That is an action which has no neighbors with which it can combine. In this case that is actually in a let take A 2. It is standing by itself. A 4 can be combined with A 4. This also A 4 A 1 A 3. So, it can combine with anything. So, A 2 has to have all four conditions specified. So, that is normally in Karnaugh map reduction you should start with the most unpromising box. Which cannot be combined with anything.

Because otherwise there is a possibility of by mistake come using the same box in more than one combination. And so, redundant term will occur. So, the important thing to start with that. Then let us look at the next most unpromising one. A 3 has no neighbor here, no neighbor here. This also has no neighbor here no neighbor here. But this A 3 and this A 3 are neighbors. Because as I said the Karnaugh map can be thought of as wrap. Wrapped around a vadai in both directions.

So that means, the top column will the top row and the bottom row are really neighbors. So, that means I can combine this and this. And normally I circle like this another circle like this. To show that these two can be combined. And if we combine it, C 1 C 2 is No Y. C 4 is constant yet No. C 3 takes on both values N and Y. That means A 3 is represented by C 1 not C 2 and C 4 not. Because of this constant. So, that is how I get that.

So, after having combined these two A 3 then I go to A 4 has got lots of neighbors looks like you know. And A 1 has also himself lots of neighbors. So, let me just take A 1. Take A 1 these four will combine. Four combine we know that two conditions will get eliminated is not it. So, the let us get back to this these four. That means I just look at the column. In this case C 1 is constant data's. And C 2 takes on both values. So, C 2 will get eliminated.

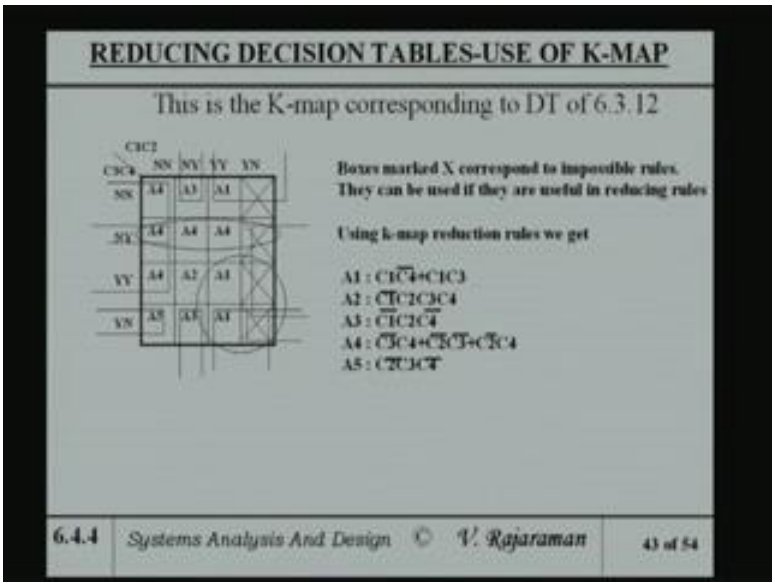
And C 4 C 3 is constant ((Refer Slide Time: 13:52)). And C 4 is Y and N. So, C 4 get eliminated. So, the A 1 now would be one. One rule would be C 1 C 3. So, A 1 will be C 1 C 3. That is one. The other A 1 is sitting here. This A 1 has got a do not care neighbor which can be combined with that. So, we can now take this four this two and this two and combine. Because we can use the same action in more than one circle or one what they call a cube in the in the Karnaugh map ((Refer Slide Time: 14:39)).

So, it can be in more than one. And so, I can combine this with this. And I will find now it is even is constant at Y. C 4 is constant at No. C 3 takes on both values. So, it is actually A 1 would be C 1 and C 4 not. So, A 4 is combined. A 1 is combined like this. Now A 4, A 4 these three are there. And there is one do not care. So, this A 4 can be combined together. And so, C 1 C 2 becomes irrelevant. Because all four possibilities are covered.

And C 3 C 4 is Y Y. That means, no N Y. That means C 3 naught C 4 is these 4 . So, I combine these four. And then, one more this A 4 is left and this A 4 is left. Now, what can I do? if I look at the neighbors, these two A 4 has got this as a neighbor. So, this can be combined. And these two A 4 can be combined with no there is no A 4 here. So, I cannot really combine there. So, these are coefficients of four box. And this A 4 I would I can combine like this.

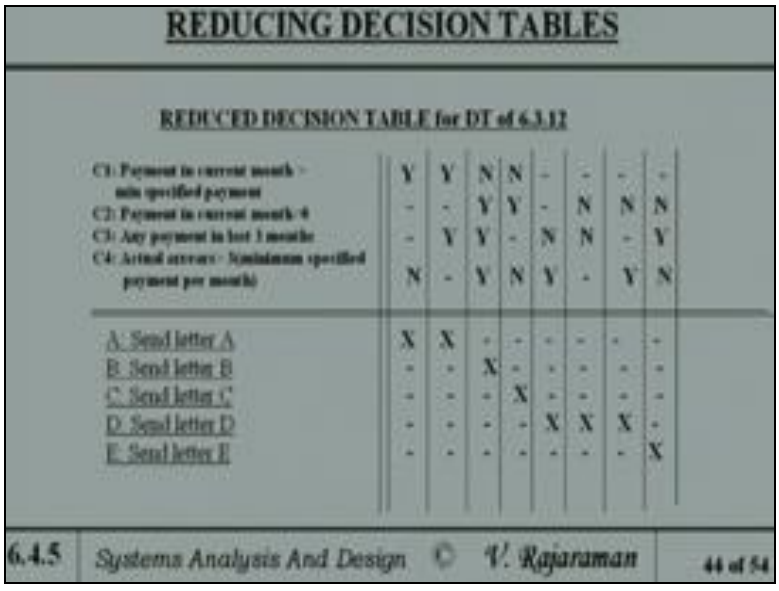
Now, that these two A 4 can be combined like this. And with do not care. Both are do not care. So, I can make a four box out of that. Two four boxes. That is one is this. This four box will again eliminate two variables. And one is C 1 not and C C not. So this itself become C two not C 1 not and C 3 not. Now, if I combine this will become C 4 and C 2 naught. Because C 1 takes on both values.

(Refer Slide Time: 17:02)



So, now, I combine. So, I will just show them in summary. See I got for C 1 A 1 C 1 C 4 naught or C 1 C 3 and A 2 C 1 naught C 2 C 3 C 4. Because all four were there. A 4 had essentially three rules. And A 5 had essentially one rule. So, this essentially what happened. And now I can map that on to the adjacent table again. That is take each one of them. And map it on and wherever variable does not occur. I put a do not care. Wherever the variable occurs I put the yes or no.

(Refer Slide Time: 17:38)



Having done this I end up with the set of rules. Which are now total number is two four six eight rules instead of twelve. From twelve it came down to eight rules. And lot of do not care conditions. That means when I write a program those conditions need not be tested. So, it will become somewhat more time effective. There are for computer programs available which will reduce or make equivalent program. Corresponding to this decision table which is optimal in some sense.

The optimality consideration in this case normally is taking now what which program. Because the same table can be mapped into different programs. This is not unique. So, which program will be the one which gives minimal time of execution. Is one criteria which normally people use. For that I should know more information. Namely what is the frequency of occurrence of each rule.

If the frequency of occurrence of each rule is given. And the time to test each condition is given. Then there are optimal optimization method does exist. To be to kind of reduce this table for equivalent program which is minimal time minimal runtime program. In fact, these programs are actually automated. In fact, provided this data is given to you. The time for testing conditions will be easy to infer from the complexity of the condition. But the frequency of occurrence of rules may not be always easy to get.

But normally what they do is with the same program like say ((Refer Slide Time: 19:49)) program or some other program our similar nature is repetitive. If run very often if it is run over a period of time then the frequencies can be found out empirically. That is the you can actually count. How many these particular rules are fired. I might say. And then count. And using the total number of experiments may Say total number of runs. I can find out an empirical probability.

The empirical probability which is ((Refer Time: 20:22)) approximation of the true probability. Depending upon the number of times I run. The larger the number of times I run. The probability will tend towards the true probability. But I think I am not going to get into this operation issue. Because little bit out of the out of our range. In terms of the fact that it will take two three lectures to explain that. Because there are many methods available. There are certain optimal method which you can guarantee are optimal.

Which use things like branch and bound and so on. Mostly operational research or research techniques and there are some heuristics. Heuristics means you get a program which is almost

minimal time. But not guaranteed to be minimal time. And that is also alright. The reason why once looks for an heuristic is that the amount of time taken to come up with a heuristic. Is ((Refer Time: 21:25)) smaller than the amount of time required to do the true optimization. And so, there is trig off again. And so, people normally look at heuristics provided one can compare.

What the time given by this heuristics says related to what is optimal. If it is close enough. Then it is and the heuristics is very simple. Then one would use a heuristic. And that is what is normally done. Mostly heuristic methods are used. what I mean by this heuristic method is a method which is approximate. Which cannot be guaranteed to be optimal. But then in a situation where the probability itself is not accurately known. Little bit of error in heuristics does not make any difference.

So in practice the optimality may have no meaning. Because you start with data itself which is not reliable there. And so, there is no. There are many situations in computer science like this. You could really find out optimal methods. But the optimization will take a long time as a computer optimization. And you also find heuristics which would not which will take less time. And depending on the situation, depending on curve what data you have available. If data itself is not available with any great accuracy.There is no point whatsoever in going where a true optimality.

Because it is very nice to have a PhD thesis with a true optimal solution. That is what gets you the PhD, but the heuristic is what is more practical in many situations. Because it does not mean that PhD's are not given for heuristics. It is given for heuristics provided heuristics is a good heuristics. And you can show that it is close to the optimal ((Refer Time: 23:24)) You get a pat in the back saying that you have come up with a much less complex problem.

It is not a N cube problem, but it is a order N problem. It is a order N cube problem. That means the solution is takes going to take much less time depending upon. See in this case the order N will depend upon the number of conditions. The because the normally the condition in this case is quite small. As I said normally you would like to break it up into smaller ones.

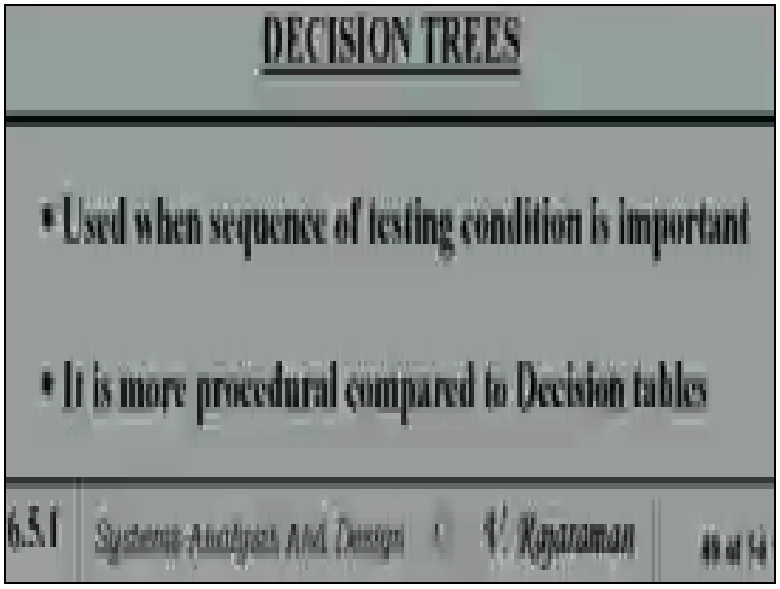
Because we have to understand it. So, that is the point. But the reason why said all this is because the advantage of with this methodology is that programs also exist. Good heuristic programs also exist. To do an automatic conversion. And your manual job is

one of making sure that. There are no contradictions incompleteness and so on. So, if you say that it is logically correct that is a very first step. Once it is logically correct then you can go ahead and write a equivalent program.

And in fact, this is a much better method than using test data. Because test data will not guarantee the correctness. It only will not show the absence of errors. It only show the presence of errors. That is the famous ((Refer Time: 24:59)) statement about the need for proof of correctness. So, this is some kind of a proof of correctness, which we have used.

But not as esoteric as a situations we are in loops and loop invariants and so on. ((Refer time: 25:18)) because the question of in esoteric or not depends upon how have taught that subject. Basic principles are fairly straight forward. In terms of the checking the correctness of program as you write programs also. And if you follow them strictly, then you will end up in situation where testing will become redundant. So, in this case testing will become redundant.

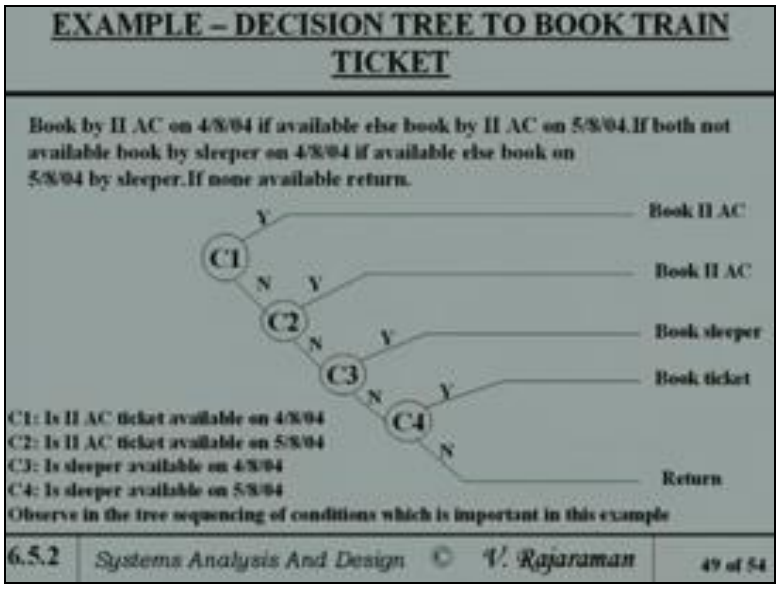
(Refer Slide Time: 25:46)



If u follow the rules very carefully. Now, I really completed the part related to decision tables. Now, I will go to the next part of the talk called decision trees. Decision trees is somewhat like flowchart. It is it is a sequential testing procedure of conditions. And when sequence of testing condition becomes important. Then we use it. And because it is more procedural. That is in other words the order of testing conditions is pre specified. Unlike in the decision table there the order is not specified.

The conditions can be tested in any order. And that is how we put them in one table. But the advantage of condition being put in any order. Is what I will helps in optimizing the program. Which is going to you know which resides in the table. Let us in this case I will do a pre decision of testing inside a sequence. And it is necessary in some cases see.

(Refer Slide Time: 27:07)



Like let me give an example. An officer sends his assistant to the railway station to book a ticket in a train. And he gives following instructions. Book a second A C on for some arbitrary rate of check four eight zero four available. If it is available on that day book it. Else book second A C on next day. If both not available book the sleeper on the fourth on the earlier day. And if not available then book it for the next day by sleeper.

If sleeper is also not available for fifth then maybe there is no point in my going on the sixth. So, I decide that return come back. If you do not say what to do in that case ((Refer Slide Time: 28:00)) go to the railway station. And if I find out that the A C first second class is not available on fourth and fifth. Sleeper is not available on fourth and fifth. So, what will he do? He will just stay a standard if he is stupid, if he is a robot.

If he is intelligent he will phone back the boss. And say sir both days not available what do I do? Then the boss has to tell that you return. Or book on sixth or whatever it is. In other words the rules are given. Word statement rules are given. Normally people are not all that careful. They just they just state the last one. And in both cases do not occur.

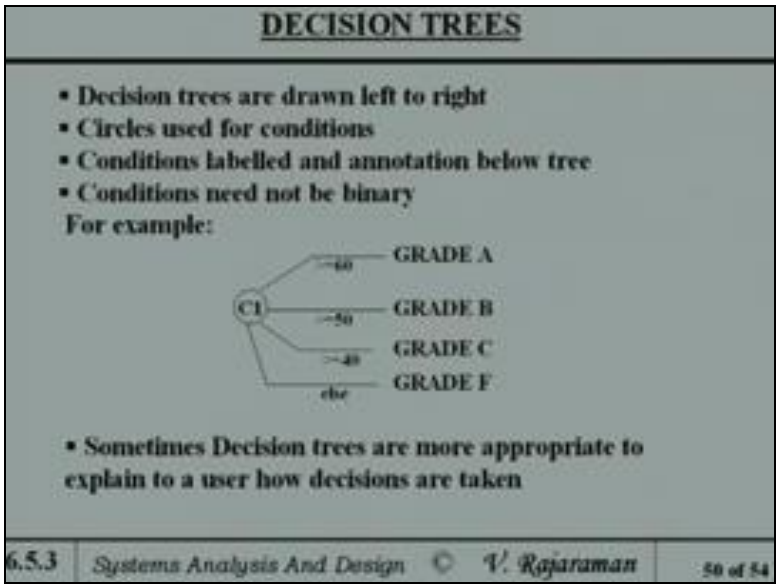
Why what do you do? They do not really say that. Because they assume that at least sleeper will be available. And they said the person to book. See that is what happens.

So, in this case then the sequencing is important. Because If A C ticket is available on fourth, I had to book it right away. C 1 so, the condition C 1 in this case which is A C second class ticket is available on four eight zero four. If it is true book second A C. If it is not true if it is not available then find out the A C second class available on next day. And if it is available book else you book for a sleeper and book sleeper. Else you go next and if it is available book for a sleeper. If nothing is available then you return.

So, this is where what I mean by the fact that this sequencing is important. Unlike if you put in a decision table that means you can turn it around in any order. And so the details not really valid in such a kind of situation. Where sequencing is important. So, sequencing is important not only in such human situations. But in many process control computers. What you try to do is to test various types of parameters of a process like temperature pressure and so on. And they normally are tested in some sequence.

And say when a sequential systems which are there to make sure that the correct sequence is followed. So, there are called sequence controllers and so on. Sequence controllers are very big area in instrumentation and desiring sequence controllers. ((Reference Slide Time: 30:34)) the point is decision trees are lot more useful. If you are desiring sequence controllers which is will occur in numbering system. Or if we are looking what word statements were sequencing is important from the point of the semantics problem. The meaning of the problem.

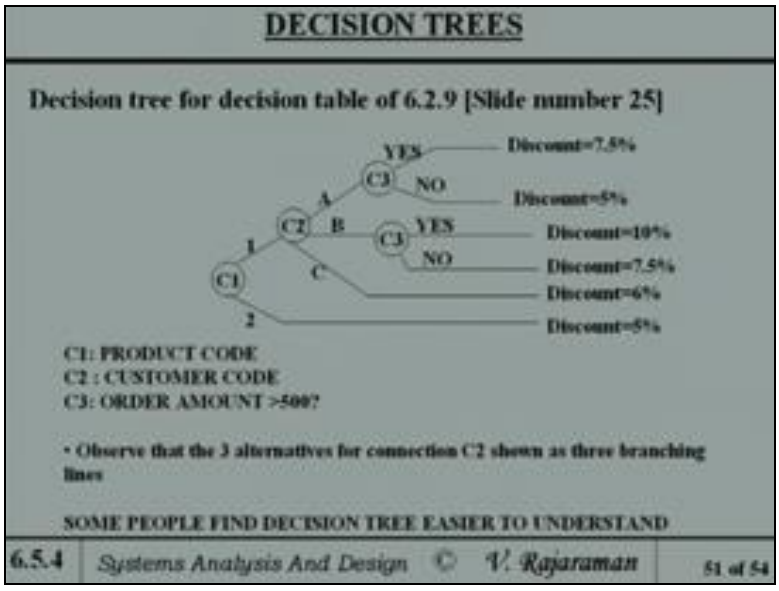
(Refer Slide Time: 30:53)



Now, the decision trees are ((Reference Slide Time: 30:58)) yes no answer. You can have a condition which has multiple possible answers. Like in this case no the I am just defining. Decision trees are drawn from left to right. Circles are used for conditions. Conditions are labeled with what the conditions are present below that. Because you want to clutter up that particular tree. And then yes or no is what is the true results continuing. But the condition need not be necessarily be binary.

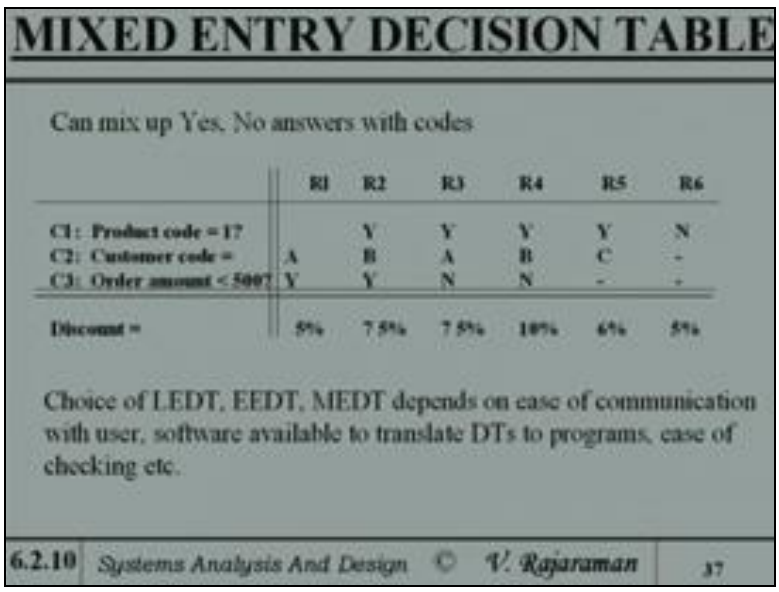
So suppose you are in an exam result. If it is greater than equal to sixty you grade A. And if it is greater than equal to fifty grade B and so on. You can give A B C, but is less than forty grade F. So, this condition got four different branches which come out of it. So, that is also allowed in decision trees. That means it is not only a binary decision tree we are looking it. It is a decision tree multiple branches can come. And each branch can lead to a different node. That can in turn lead to multiple branches.

(Refer Slide Time: 32:21)



Now, I am showing a decision tree corresponding to an earlier slide. Namely the discount rules which are there. The discount rules let us just go back and yes.

(Refer Slide Time: 32:34)

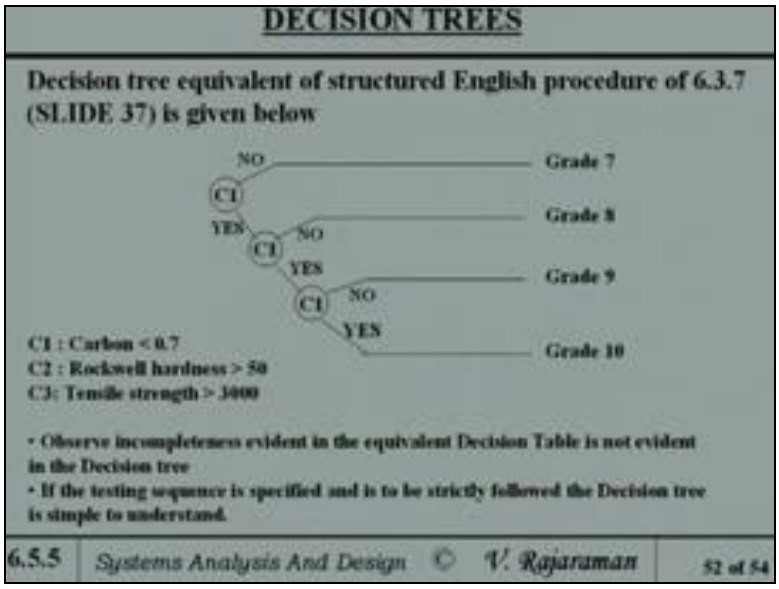


Product code one customer code A B C and order amount ((Refer Time: 32:43)) which I had. Where the actions were giving discount and then, I had a product code one or two. One in this case one yes or no. So, I am assuming one or two can be used. And there are three possible customers A B C. And order amount is the third one. So, this is what converted tree.

If product code is two then discount is five percent. That is what that is what the I think I have urged you to go back and put the details in front of you, and go through it. I have a problem in the sense that I cannot accommodate in one PPT. So if it is one then you find out customer code A B C. And then, find out if the condition C three is about how much of order amount is there. The order is above five hundred. Discount seven one seven and half percent. If it is below then discount five percent.

Table is now reflected as a tree. Now apiary of this decided to test condition C 1 C 2 C 3 in that order. But it need not be as per the ((Refer Time: 34:02)) rules. I can In fact, start with any one of them. But this looks more logical. And so, it is easy to explain also. That's why I did it this way. Thus the point the reason why I put this is that. The tree can have is an not necessarily just two outputs from any circle. You can have multiple outputs.

(Refer Slide Time: 34:29)

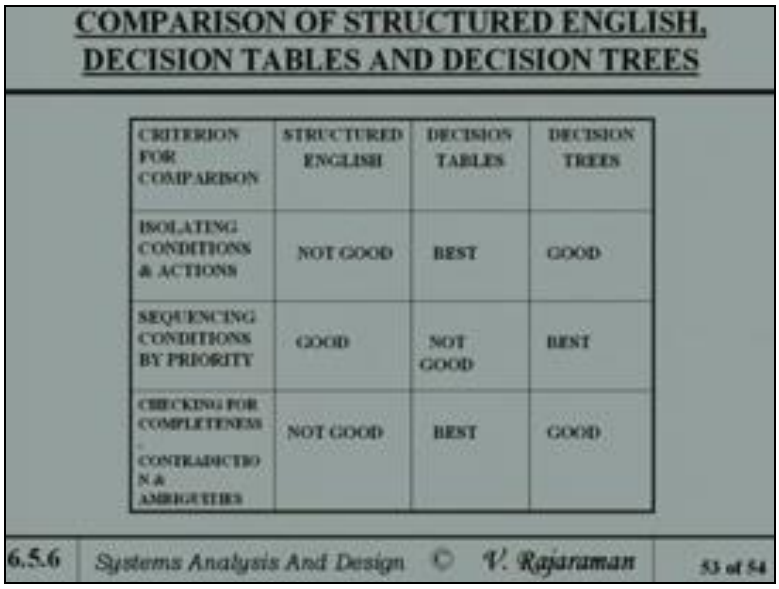


And that is what shows here. Some places there are two outputs. And some are with, there are. So, the other one which you had is that poor audio carbon content less than point seven. Rockwell hardness greater than five hundred. Tensile strength greater than three thousand. We had a decision table there. Where we found out that if the YES NO YES. You know the YES YES YES and YES NO and YES YES NO and YES is followed. Then there are grades given.

But suppose it happens to be say the answer C 1 is NO. Then the you know the C 2. I think there is a error in this. This should be second circle should be C 2. And third circle should be C 3. There are three different conditions which are being tested C 1 C 2 and C 3. And but here again the fact that there are certain ((Refer Time: 35:41)) conditions. In this case YES YES YES is specified. But suppose NO YES YES is there. There is no C 1 NO C 2 YES and C 3 YES.

NO YES YES is specified then what do I do? Is clear in the decision table, but it is not clear here. So the completeness of the decision table can be checked very easily. Whereas this because like a flowchart. I cannot check completeness. Because having started test with say condition C 1. Then the only other alternative is to test condition C 2 next. Because it is sequencing is there. So, C 1 C 2 C 3 in that order. And So, it cannot really find out the error in the decision tree.

(Refer Slide Time: 36:28)



Now let us compare the three methods. We had three different methods of specifying processes. We started this module with power specification. And we looked at structured English. Then we looked at decision tables. Then we looked at decision trees. So, there are three different ways of actually specifying processes. And the question which arises is that. Which one would I use, under what conditions. And what are the advantages or good points and bad points about these particular methods.

Now, the criteria make compared are isolating conditions and actions. Which one of them isolates conditions and actions better. Which one is good at sequencing conditions by priority. Which one is good at checking completeness contradiction and ambiguities. In other words the logical correctness of a set of conditions. Now, structured English is not good in isolating actions and conditions. Because there are mixed up in that process statement.

Some conditions are checked. And then, some actions are taken. And some other conditions are checked and so on. And it is good for sequencing condition by priority. Because when you write the process specification in some sequence. By implication you are saying that certain conditions are to be tested in a certain order. And sequencing details are no good at all. Because we are assuming that there no sequencing involved.

As certain conditions are considered. Only place where sequencing is become important in the decision table in terms of linked decision tables. When you have a link table where one table leads to another table. Then the order of linking become important. So, there could be some situations where ever whole set of complex rules are given. You parse a ((Refer Time: 38:46)) in other words you Parsing means kind of reducing it to set of consistent set of rules.

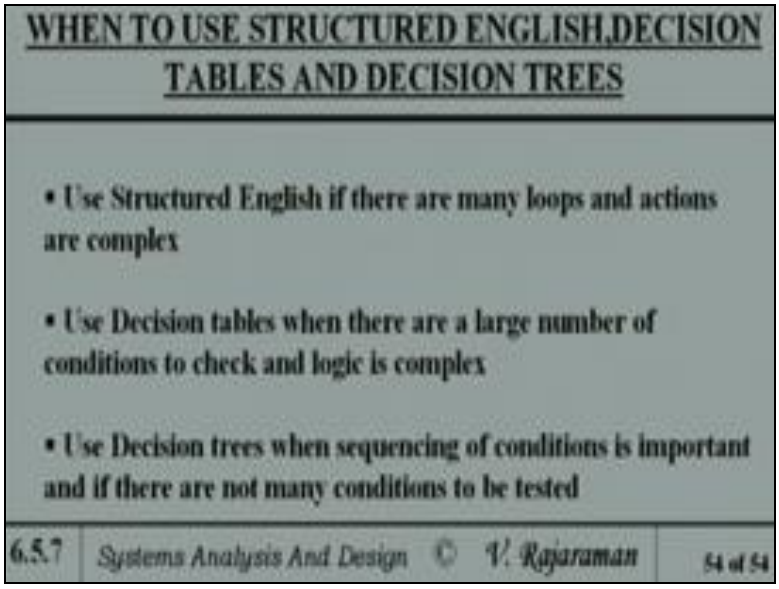
And so when you do a parsing of this consistent set of rules then you come up with a link table. Because if you did not parse it. Then the number of conditions in one table can become in order ((Refer Time: 39:17)). And the advantage of a table in terms of completeness checking and ambiguity checking and contradiction checking and all that will be difficult. And it will become (Refer Time: 39:33).

That is the reason why we kind of parse it to make smaller tables. But not more than five or six conditions in each table. But then sequencing is implied when you have a because those three those tables are in a tree now. One after the other in some order. Isolating conditions and actions details are best. And they are not good for sequencing. And for checking completeness contradictions and so on. It is the best methodology.

Decision trees are good for isolating conditions and actions. They are best for sequencing conditions. But they are not very good for completeness checking and so on. So, it somewhat similar to the more concise and process specification. But today's it has got disadvantages. So, depends on the test some managers prefer decision trees. Because

they can actually look at the picture of what decisions are being taken in some order and so on.

(Refer Slide Time: 40:30)

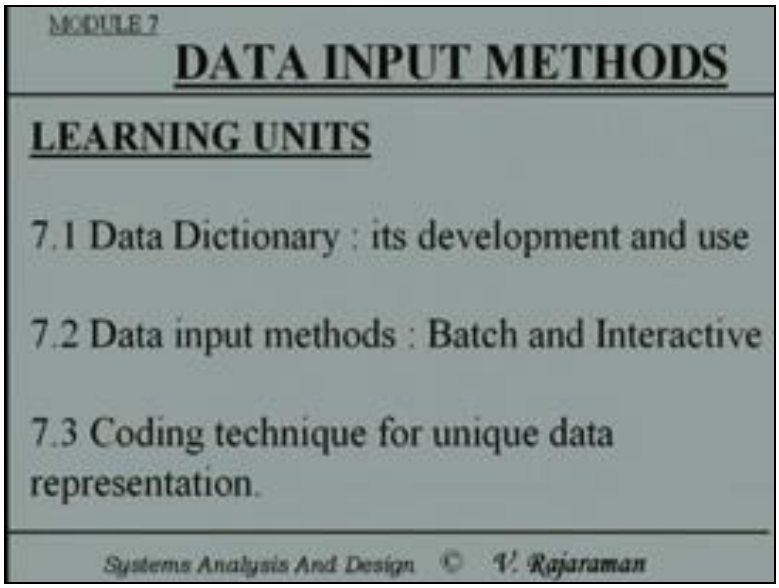


But it depends there it is purely a equation ((Refer Time: 40:47)). There would constrain correct proof of correctness and so on. And if you have a complex set of rules nothing like details. You had to use them. Very complex set of rules and you want to really be sure. Or prove the correctness. So, that is the reason I spent a fair amount of time. In the lecture because many books kind of gloss over it. They do not talk about the aspects related to the proof of correctness.

Which I believe is important. Because rules are becoming complex all the time. So, the final conclusion you might say are suggestion to you is that. Use structured English if there are many loops and actions are complex. Looping is very easily represented in a process specification. While loop and do loop and things like that. Use decision tables when there are large number of conditions to check. And the logic is complex.

Use decision trees when sequencing conditions is important. And there are not too many conditions to be tested. And then use this. So, this is essentially the final advice you might say. At the end of this module in terms of three different techniques we have described. And went to use each one of them. Now, having completed this module. Now start on the next module.

(Refer Slide Time: 42: 30)

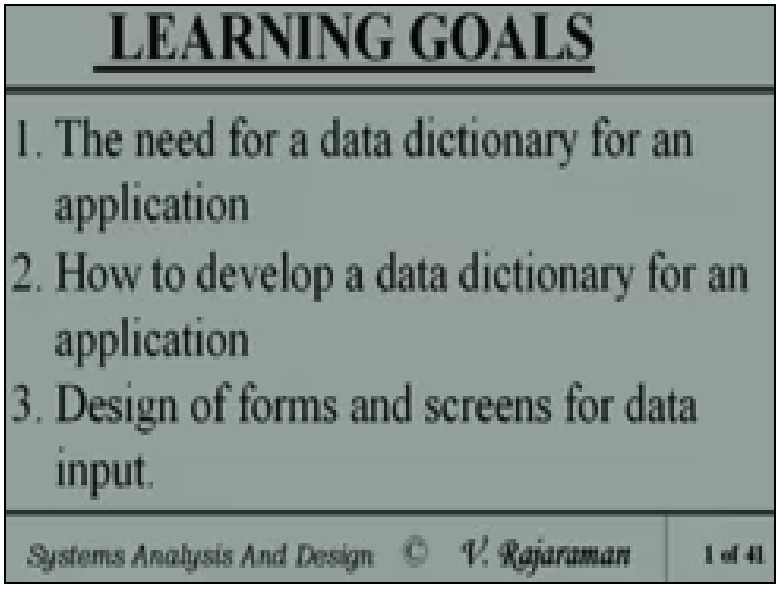


Next module is a important one. Which is on data input methods. And I am going to divide it up my discussion into a number of different parts. One is I am going to talk about what is known as data dictionary. How it is developed and used. And then, about data input methods. How to input data. Batch data as far as interactive data. when I say batch input like it is like. You know the few days ago the results of, you know CBSE and other exam were announced.

Where the number of candidates who take the exam is runs into lakhs. Two or Two lakhs three lakhs four lakhs and so on. State board runs into tens of lakhs. In that case it is a batch which we use. And batches taken together. And batch whether already you might say. The marks sheets which are entered by manually are given to a data entry operator. Enter the data of the data entry machine. That you what to mean by batch. And interactive means there are situations like whereas point of sales.

In other words in a super market when you purchase some items. The girl at the counter will just enter the name of the article. And amount quantity you bought price per unit and the total amount. And then it will total at the end. So, this is interactive while she enters it for each item as it comes. And then gives you immediately a bill. That is interactive. So, the methods are revised for batch entry are somewhat different from with interactive entry.

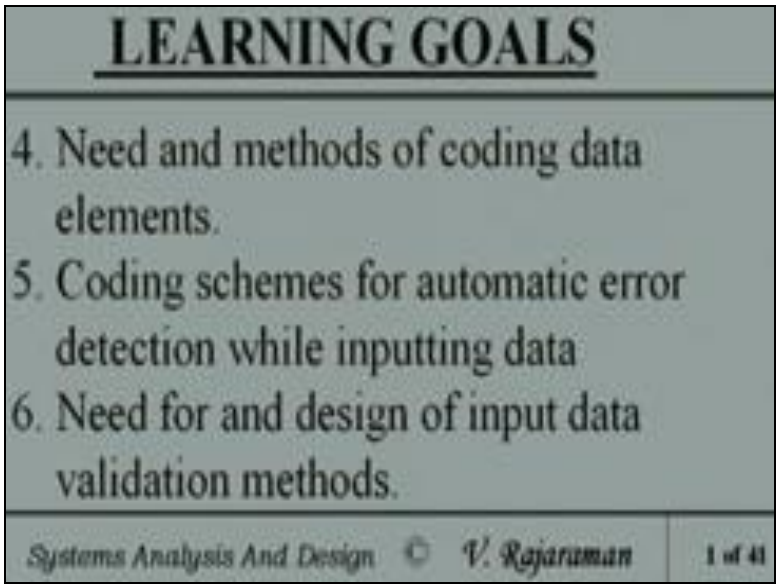
(Refer Slide Time: 44:42)



Because interactive entry you need to have feedback immediately. ((Refer Time: 44:33)) error was made. And coding techniques which is very important for the innerving uniqueness of data. Particular key data representation. So, these are three major topics I am going to cover. The goals of our these talks that why do you in other words we have to ask the question. Why do we need data dictionary for an application. And how do we develop data dictionary for an application.

And in a part of the how to you have learn about design of forms and screens for data input. Data dictionary is a raw material from which I start. And that raw material is used in the data input. And So, design of forms and screens. Forms are normally used for batch. And screens are used for interactive. Need and methods of you know how to code data elements. Particular key data elements.

(Refer Slide Time: 45:36)



Why do we code it, and what are the coding schemes which are there for automatic detection of errors. Because detection detection of errors while we enter key code is extremely important. And ((Refer Time: 46:00)) is made in the entrance of a entry of a key code. That can lead to error in terms of the rest of the system. So, automatic methods of checking the any error in the key entry is normally used. And one would very must have like to have a system where it also corrects.

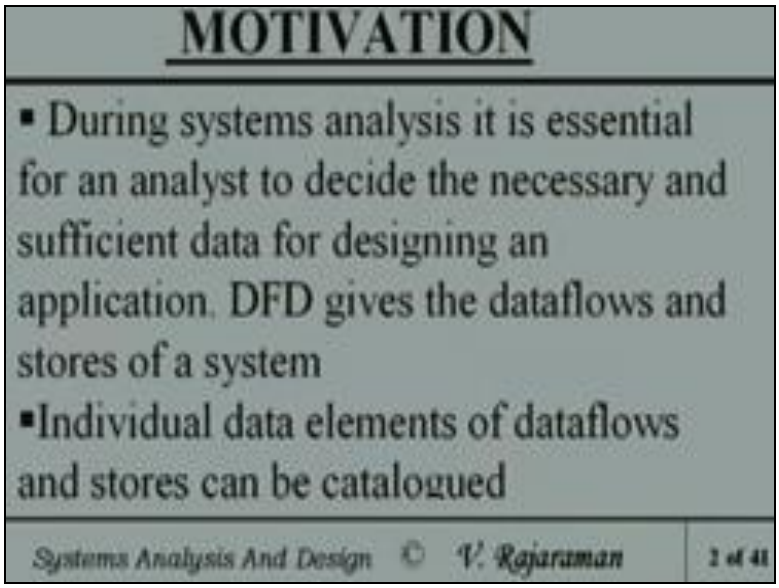
Review enter the wrong one it detects and corrects. But it is possible. It is not impossible to do that, but it turns out that if you want to do detecting and correcting. The number of redundant digits you have to use to be able to correct. Becomes quite long. And so, total length of field becomes long. And so, the possibilities that the correction ((Refer Time: 46:51)) also somebody makes a mistake arises. So, this system must be such that if the error is made in any place where is the correction digit or a two digit still it should be corrected.

So, that is what makes it very complicated. In binary systems for instance. There are error detecting and correcting codes. Having code is a is a common code use for having parity check codes are used for error detecting. Because you know the number of ones are either odd or even or you just count. And parity is used all the time in digital computer design. And there also there are error detecting and correcting methods.

Where if the errors made more than one bit. Then you automatically correct it. Those are used in certain kind of situations automatically by the computer. Particularly in areas like you know long distance communication. Where a set of bits are sent and there is lot of noise in the in the atmosphere. Then when the word is received they automatically correct it and before using it ((Refer Time: 48:12)). So, there are situations particular secret codes in ciphers and things like that.

Where error detecting and correcting become important. So in this case also one could detect except that this is not binary. The codes are normally alphanumeric that means, they have got letters and digits also in it. And because a little bit more complicated. But it is possible. It is in theory it is possible. And so, the need to design input data validation methods. That is apart from see coding schemes. When you enter input data you have to do validation of that data and that is very important.

(Refer Slide Time: 48:54)



And the validation important is important again. Because of the sheer volume of data being entered. In fact, you know yesterday my neighbor's girl came crying and said you know I got my mark sheet. And I got in physics thirty six out of hundred. And mathematics I got hundred and twenty six out of hundred. So obviously, that is a data entry error which has been made.

Nobody can get more than one twenty six out of one hundred. And maybe thirty six is very low because her marks in other subjects are quite reasonably high. In which case in

this also data entry error. Now, the point really is such an error when it occurs. It comes to the customer, because in this case the poor girl. Who as to go back and ask for revaluation stuff like that and half of some money. But in the case of customers for whom you are doing this work.

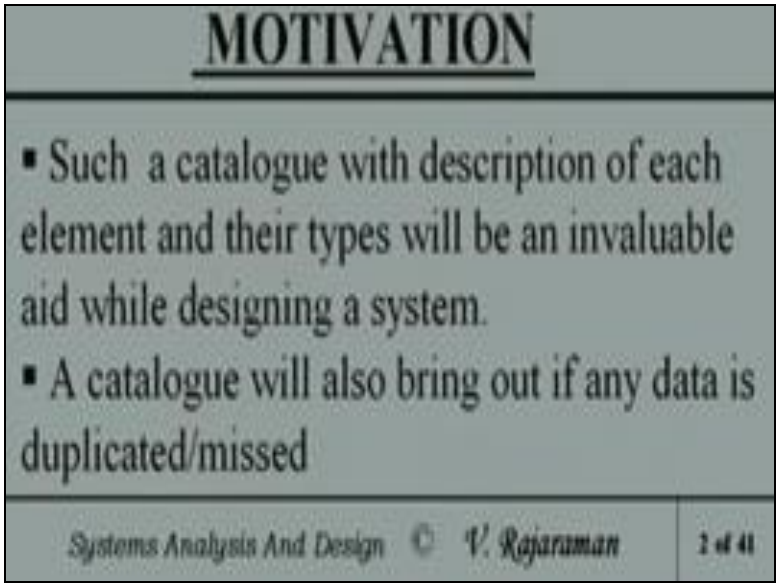
If you make some such stupid errors. He is going to kind of terminate your contract. That deal your data entry is so bad that I cannot trust your system. So, it is extremely important in while inputting large values of data. To have automatic methods of checking. And eliminating such stupid situation like getting marks above hundred when it is not possible. Or getting very low marks when marks in the other areas are reasonably high. So, these things should be find out right at the time of entry.

So, that they can be removed before the become too dangerous. In the too dangerous in the sense that. They are used later on somewhere else. Then it can become dangerous. In this case of course, as far as girl is concerned it is very dangerous for her. Because she has a sleepless night. I mean sleepless nights and I have to give money for revaluation and so on. Which is not good at all.

If this is this is really a bad kind of a system design. Which has been done by whichever board does this. They do not it in a hurry. They have not done a proper job. This is occurs is not a isolated case. Year after year this seem to be occurring. So, I think it is very important responsible. I think responsible systems analyst and system designer. Should really make sure that such stupid others do not ever occur. Because the entire credibility of computer base systems will be lost.

Because when public will start saying that ((Refer Time: 51:36)) low marks. They will say the computer gave me no marks. So, that effectively is the reason why the credibility about computerization will lost. It is very important. It is social responsibility on the part of all of us who do this system design. And systems analysis to make sure that such things never occur. So, it is a during systems analysis it is essential for an analyst to decide the necessary and sufficient data for designing an application.

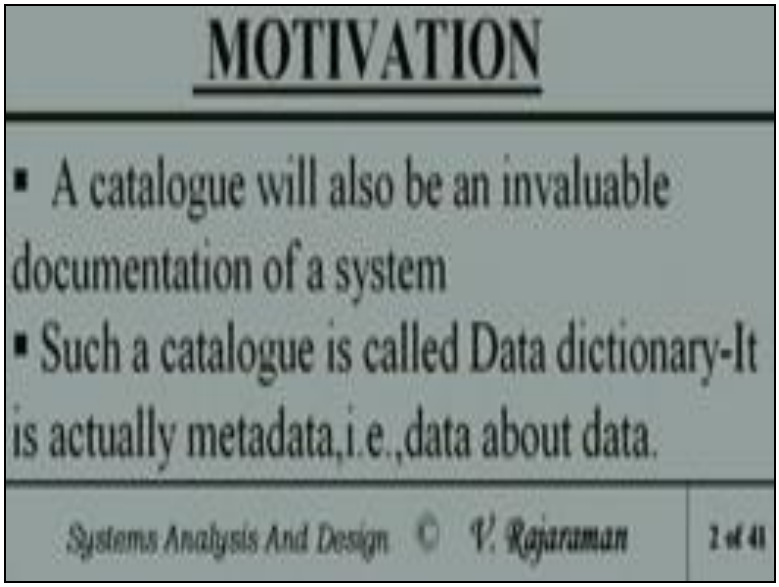
(Refer Slide Time: 52:30)



So this is obtained on the dataflow diagram. Dataflow diagram tells you what data entire system. What are processed, what are in files what are output and so on. And individual data elements of data flows and stores can be catalogued. And this catalogue is a description of all the elements. And it is a tremendous documentation name. We can detect at this time any duplicated data or missed data and things like that. And it is a very good documentation. And it is very useful for designing system.

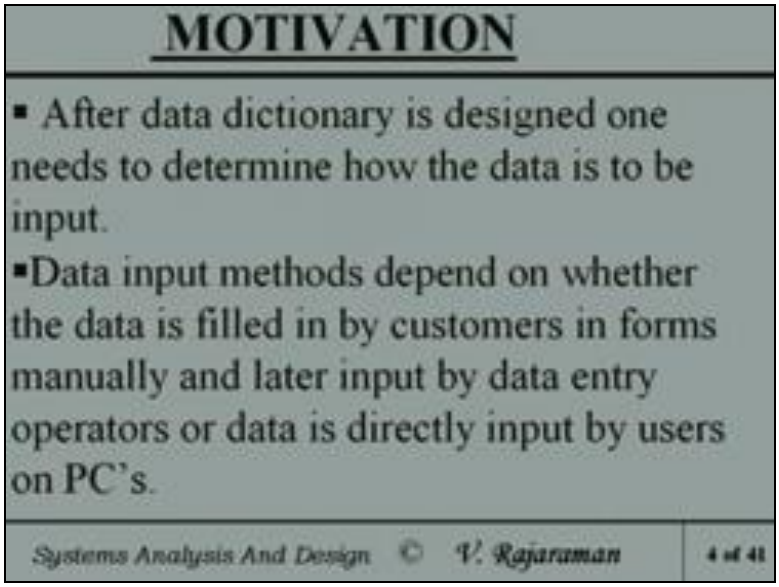
Later on if somebody is going to modify your system. ((Refer Time: 52:56)) useful. Because at a glance the meaning of that data will be in the dictionary. Some data item you would have used. Unless the meaning is obvious to a person who is going to maintain a program written by somebody else. Then you will have great amount of problem to maintain that program. And as a in computer computing maintenance is one of the major issues.

(Refer Slide Time: 53:36)



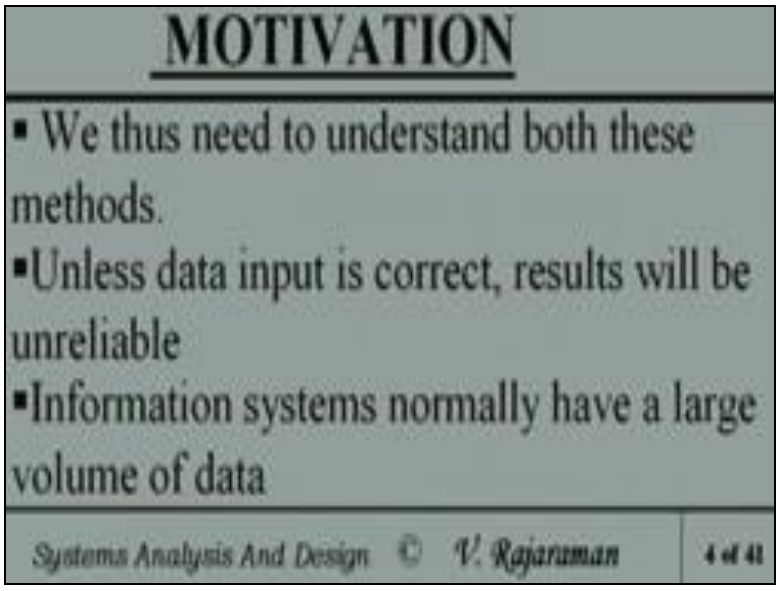
In that companies make money and maintenance year after year. So, the maintenance is normally done by a person, who did not develop the system. So, developer as a responsibility to catalogue. So, it is an invaluable documentation name. It such a catalogue is called as data dictionary. It is actually a metadata. That is data about data. Normally it is called metadata. That is the dictionary tells about what data is used and what is their meaning. And also more details about the data. So, it is data about data.

(Refer Slide Time: 54:00)



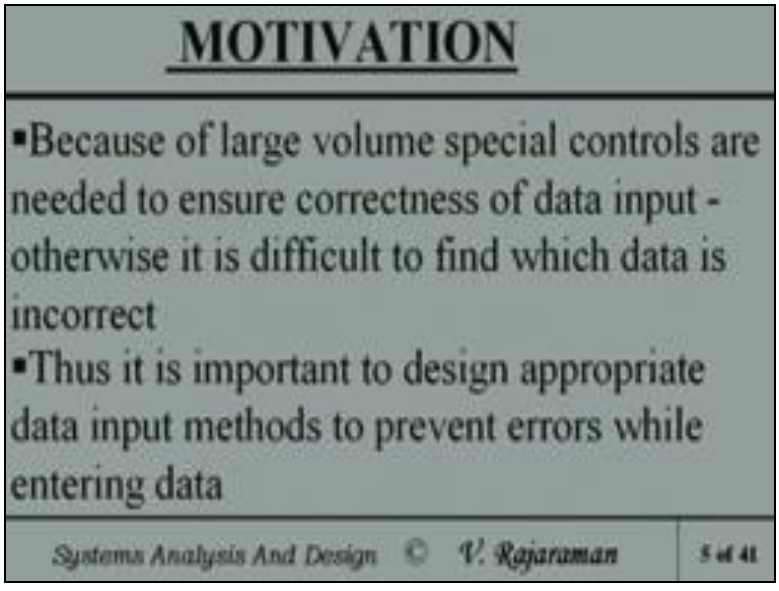
After data dictionary is designed one needs to determine how the data is to be input. The input methods we looked at. Data input methods depend upon whether it is form filled by customer. Or in this case exams by clerks. Or the input manually by operators interactive.

(Refer Slide Time: 54:26)



We thus need to understand both these methods. Both the batch inputs as well as interactive inputs. Unless data is correct results will be unreliable that is obvious. And information systems normally have a large volume of data.

(Refer Slide Time: 54:44)

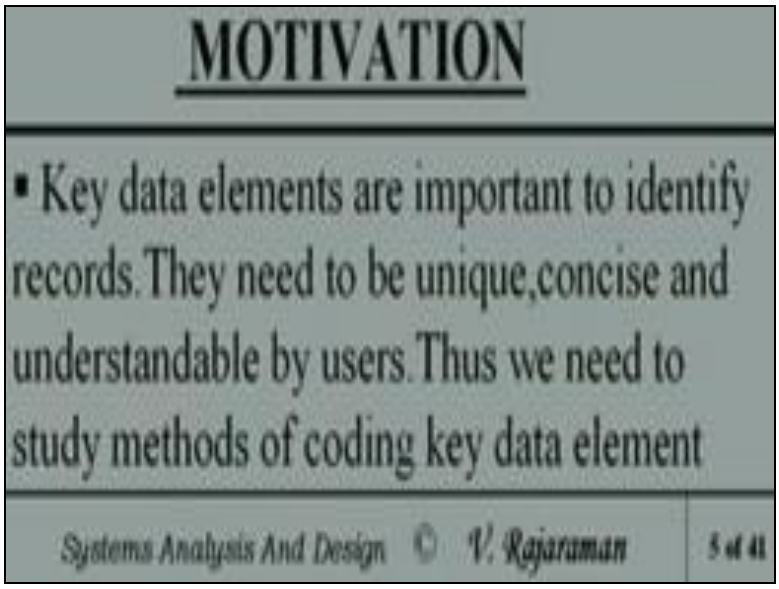


This is what the reason why we need to have a simple method of checking. Controlling the input data. Because of large volume special controls are needed. To ensure correctness of data input. Otherwise it is difficult to find out which data is incorrect after the event. See we have processed four lakhs answer. You know four lakhs entries you

made. And then, you suddenly found out some error. Nobody can go through manually all that four lakhs to find out, where the error occurred.

And say it is very important. To have methods the tracking of errors and original error can be found out. So, this is tracking extremely important. Important to design appropriate data input methods to prevent errors during entering data. Now, ((Refer Time: 55:38)) tracking if any errors actually occurred. Now, you try to prevent it. But sometimes you know in spite of all your best efforts some errors would enter.

(Refer Slide Time: 56:06)

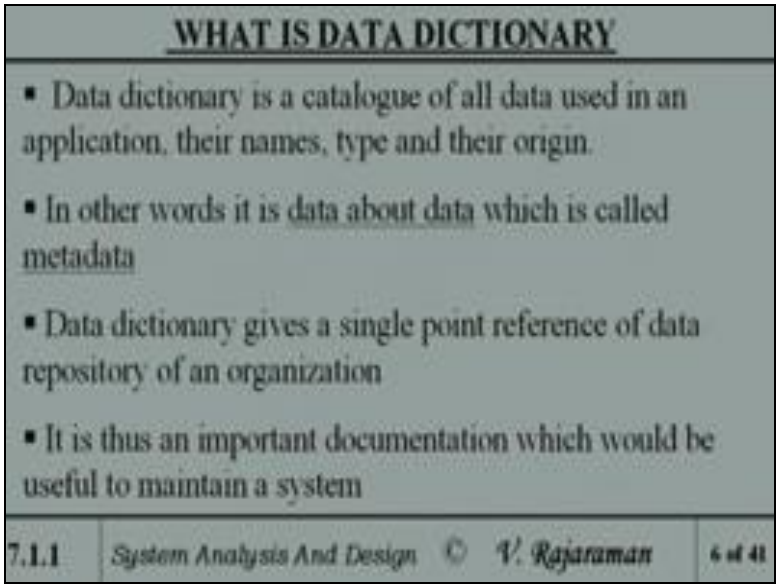


And in this case if the error entered you have to be able to track where it occurred. And you have to be able to change it. The tracking is as important as detecting. Key data elements are extremely important to identify records. Like you know for instance they if you are looking at student result examination results. The key element is roll number. Roll number is the key element. And if any error is made in the roll number and two roll number were equal. Then obviously, you have a great difficulty at the last moment.

And so, two roll numbers cannot be equal obviously. And they should be unique. And the uniqueness also, it kind of is an item which makes a difference. In terms of you know base on that roll number. You are going to print out the mark sheet. If the wrong roll number is there the wrong mark sheet will be printed. So, there is ((Refer Time: 56:57)) very important to be able to make sure that key data element no errors occur.

So, you have to quote them in such a way. That they are unique, concise understandable by users and also. They should have some correction probability. In other words if you can at least detection. I am not correction. You should be able to detect any error. In key data entered. In real number particularly. If first of all you try to prevent it by the proper design. And after doing a proper design and preventing it.

(Refer Slide Time: 57:52)



You if you ((Refer Time: 57:35)) happens during data entry. You have to be able to point out that error has occurred that stage itself. Not at the end of the game. So, that is also important. So, detecting is important. This is the reason. So, now the first topic. Is about what is data dictionary. And data dictionary essentially is a catalogue of data used in applications. There names types and their where they occur. As I said they are data about data.

Data dictionary gives a single point reference of data repository of an organization. It is thus an important documentation which would be useful to maintain a system. This is where I said them earlier. But I am essentially repeating them**.** ((Refer Time: 58:32)) and this is where I stopped. I will continue next time from this point onwards.