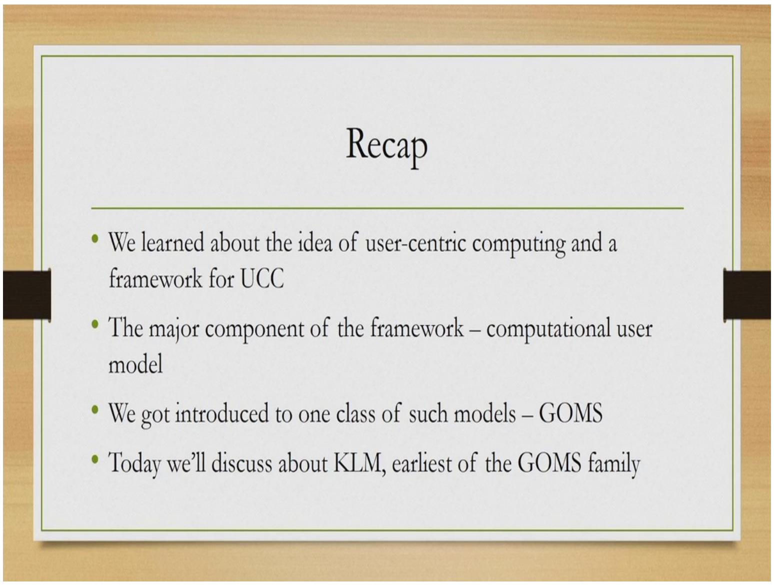
User-Centric Computing for Human-Computer Interaction
Prof. Samit Bhattacharya
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture – 14
Keystroke-Level Model (KLM)

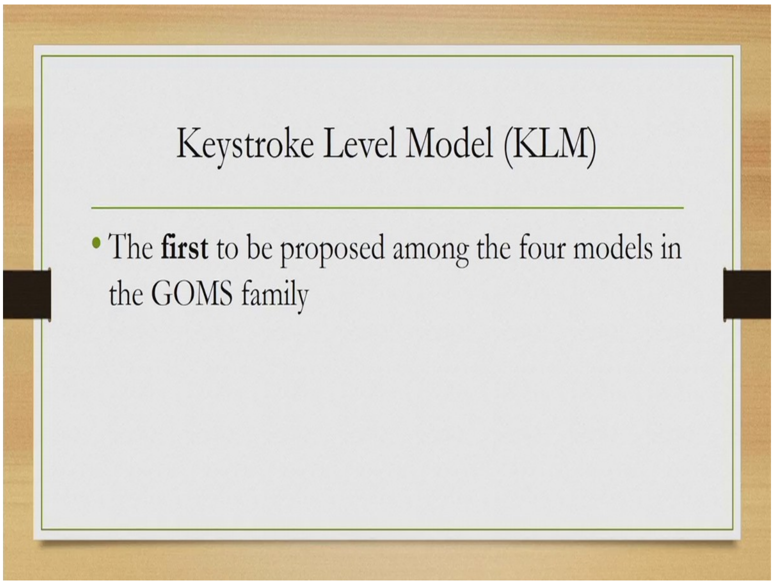
Hello and welcome to the 14th lecture in the course User-Centric Computing for Human Computer Interaction. So, like we did in our previous lectures so, we will start by recollecting what we have introduced in the previous lectures.

(Refer Slide Time: 00:57)



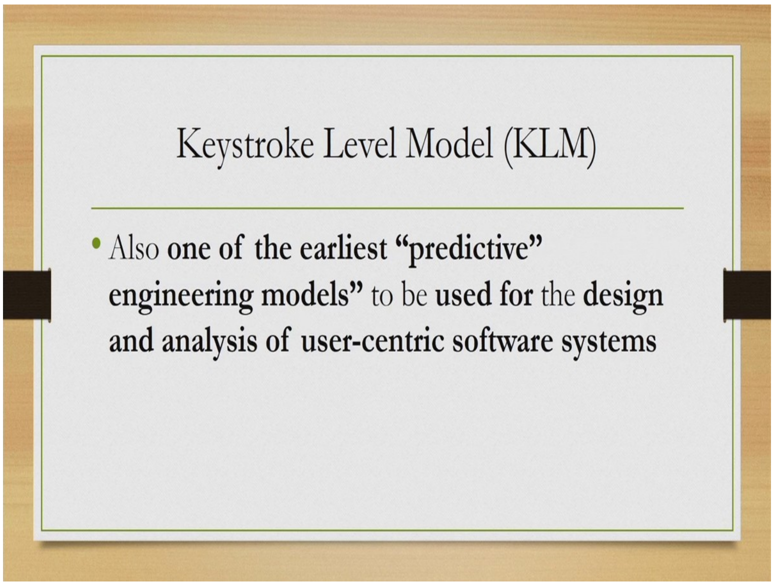
So, if you may recollect in the last lecture we introduced one family of models that is called the GOMS models. This is one class of the predictive engineering models that we introduced in the last lecture.

In the introduction, we discussed about the basic understanding of the models what they assume and the general characteristics of those models. Now, there are four such models if you may recollect one was KLM, the others are CMN GOMS, N GOMS L and CPM GOMS. Among these four as we mentioned earlier we will discuss about two of the models namely the KLM and the CMN GOMS. (Refer Slide Time: 02:11)



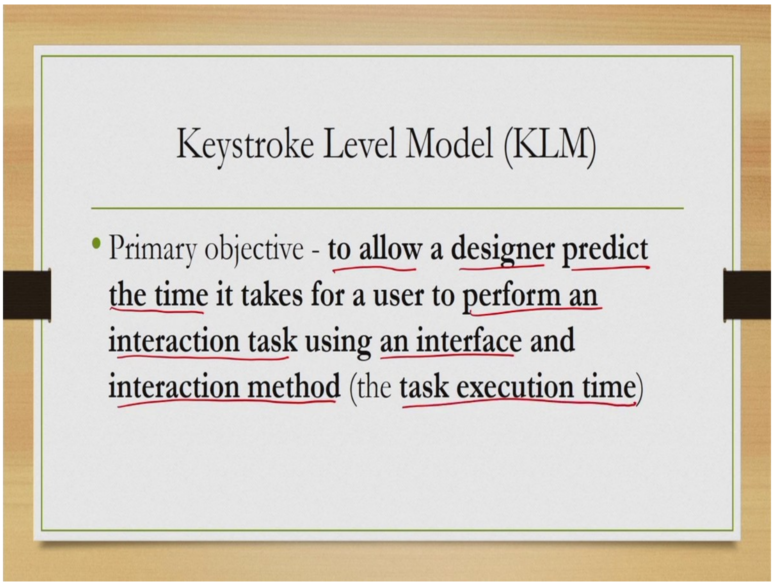
So, today we will discuss about the KLM or Keystroke Level Model which is one of the models in that GOMS family.

(Refer Slide Time: 02:15)



The KLM is the earliest of the models that are there in the family. It is one of the earliest predictive engineering model to be used for the design and analysis of user-centric system. So, KLM has the distinction of being the earliest model in the GOMS family of models as well as being amongst the earliest predictive engineering models that are developed for design and analysis of user-centric systems.

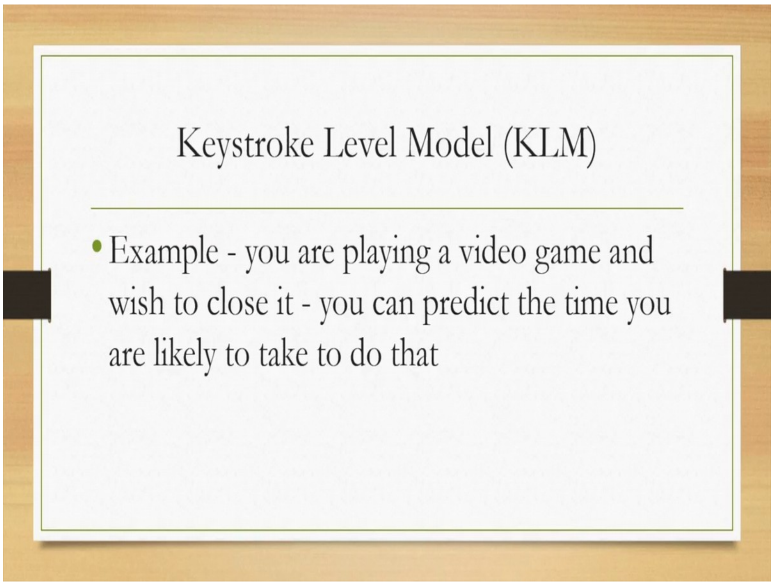
(Refer Slide Time: 02:43)



So, what is the primary objective of this model? So, remember we are discussing about predictive engineering model; that means, these models are supposed to predict something. In the generic definition, we talked about predicting the next state of the user, but that is the most generic way of understanding it. Now, this next state can be understood in terms of some explicit state variables or in terms of some numbers, some values that are indirectly representative of the state.

So, in the case of KLM or Keystroke Level Model the primary objective is to allow a designer predict the time it takes for a user to perform and interaction task using an interface and an interaction method. So, in other words it allows the designer to predict task execution time. So, if you are a designer of an interactive system and your system supports setup tasks that the user can perform with various features that you have designed for your interface, now, with KLM you can actually predict the time that it takes for a user to perform these tasks.

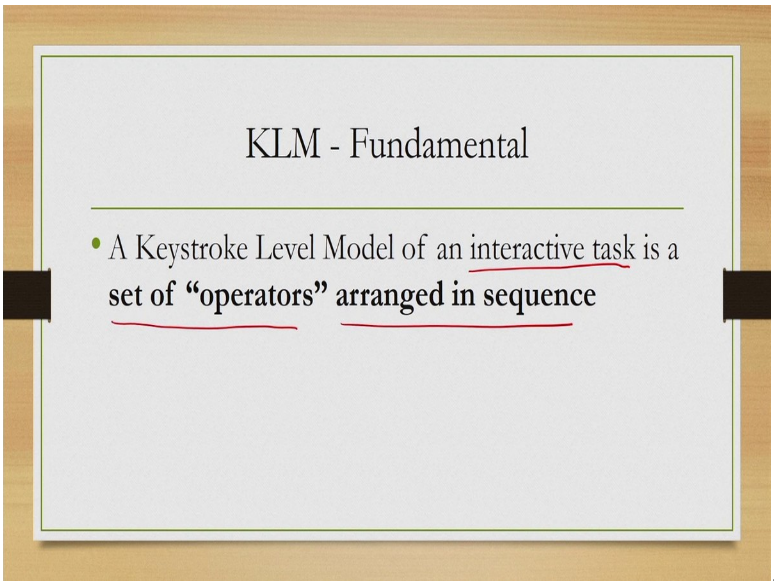
(Refer Slide Time: 04:11)



As an example, suppose you are playing a video game or you are simply watching a video and wish to close it. With a KLM you can actually predict the time you are likely to take to do that.

So, watching a video or playing a video game is essentially performing a task with an interface and if you have a KLM for the task, then you can actually predict the total time that it takes to perform the task in this case how much time it will take to close the video or close the game.

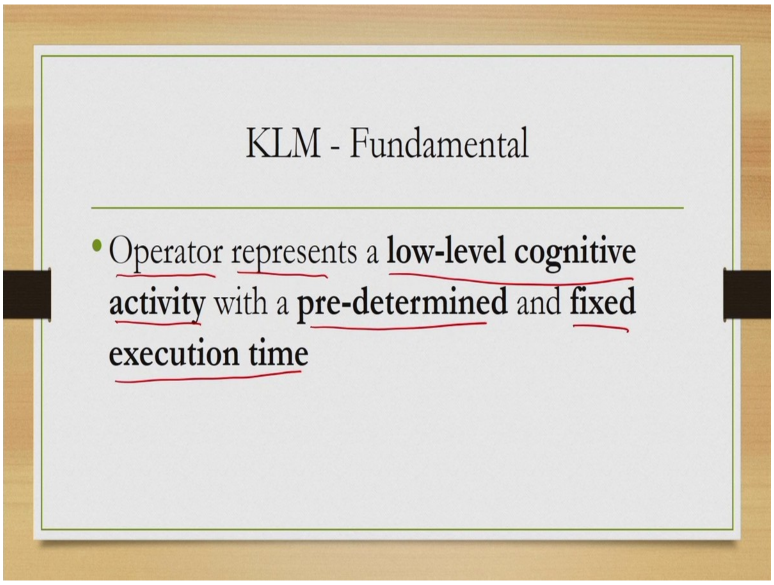
(Refer Slide Time: 04:55)



.

Now the fundamental concept in a KLM model is that the model is a set of operators arranged in sequence. So, when we are talking of a KLM or keystroke level model we are talking of the model of an interactive task and the model is nothing, but a set of operators that are arranged in a sequence.

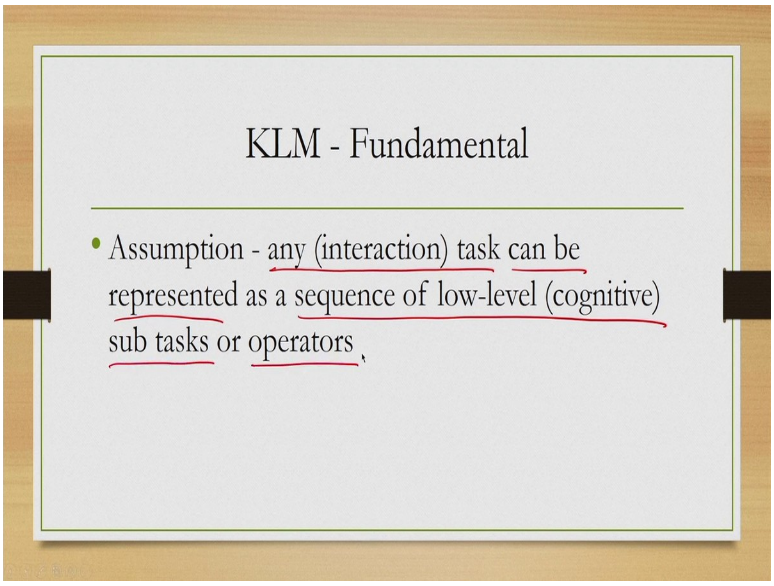
(Refer Slide Time: 05:23)



So, when you talk of the term operator, what you mean? An operator essentially represents low level cognitive activity with a predetermined and fixed execution time. Now, here it needs some clarification. So, when we think of cognition as we have discussed in an earlier lecture cognition is essentially a process. Now, in this process we can view it as a set of states, each state is a cognitive state and there are ways to transition from one state to another.

Now, the term operator when we are using this term operator as it is used in a mathematical domain, it essentially indicates an operator that works to help us transition from one state to another. So, that is the implicit meaning, but for simple understanding we can think of it as a low level cognitive task which has a predetermined and fixed execution time.

(Refer Slide Time: 06:47)



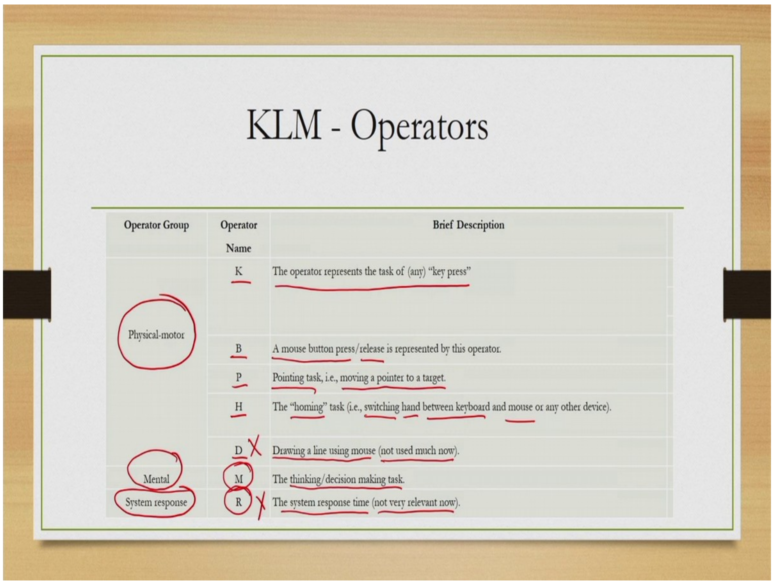
And, the fundamental assumption in KLM is that an interaction task can be represented as a sequence of low-level cognitive subtasks or operators. So, when we are talking of a task interaction task and if you may recollect our earlier discussion the task involves both the computer as well as human the user. So, when you say the task involves the user essentially what you mean is that the task involves our cognitive process or cognition.

Now, when we want to model these task essentially what we try to model is modeling the cognitive behavior and KLM assumes that such behavior can be modeled by representing the cognition as a sequence of low level cognitive tasks or operators. So, that is the fundamental assumption behind the idea of KLM or Keystroke Level Model. Now, what are these operators? When we talk of operators we gave a generic definition that these are low level cognitive tasks.

(Refer Slide Time: 08:01)

Now, in KLM total seven operators were defined. And, these operators were organized in the form of three groups. So, there are seven operators and they are organized into three groups. So, let us see what are these operators and what are these groups.

(Refer Slide Time: 08:19)



So, in this table as you can see we have the three groups: one group is called physical motor operators, one group is called mental operators and one group is called system response operators.

Now, in the physical motor group there are five operators – K, B, P, H and D. Now, these are the names given to these operators and their significance is the operator K denotes the task of any key press. So, whenever you are pressing a key essentially it is assumed that you are using the operator K, B represents a mouse button press or release. Note that when we select something we click on a button that involves both press as well as release. With D we represent either of these either press or release. So, together if you want to represent a click then we require two Bs – B, B.

Then, P indicates pointing task such as moving a pointer to a target. H is called the homing task; in other words it means switching your hand between a keyboard and a mouse or any other device. So, when you are interacting with the help of multiple devices such as mouse and keyboard together say for example, then if you switch between these two devices for an interaction task then you are said to be performing a homing task. And, when you switch that is captured in the form of these operator H.

And, finally, D represents drawing of a line using mouse which was actually relevant at the time when KLM was proposed. Remember that it was proposed in the early 1980s, in fact, the first paper came out in 1980 and it found its use in the early days of GUIs and PCs that time the operator D used to have some significance where it represented a drawing of a line using mouse, but this operator is no longer used much in analyzing or modeling modern day interactive tasks.

The second group is the mental group of operators. Under these group there is only one operator M. Now, this M indicates or captures the thinking or decision making task. So, whenever you need to take a decision, or whenever you need to think so, the entire thing is captured in terms of a single operator that is M.

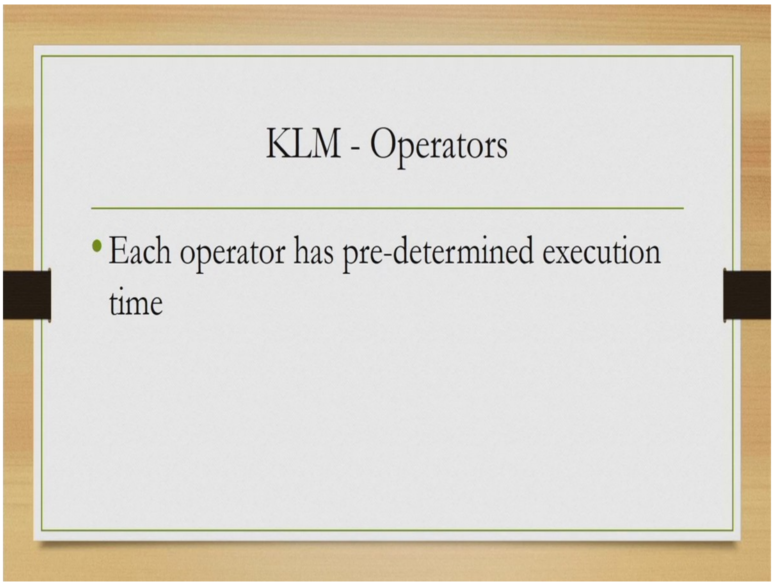
And, finally, there is this system response group which again has only one operator R which refers to the system response time. This is again another operator that had its relevance in the early days of the use of this model that is in the 1980s when the computers used to be slow and after you perform some input action there use to be some delay in getting the response getting the output due to the low configuration of the computer, but that is no longer relevant nowadays and hence this operator is not very much used nowadays.

So, to recap there are three groups physical motor groups indicates operators that are related to our physical and motor activities – namely K for keystroke operation, B for mouse button press or release operation, P for pointing tasks, H for homing task that is switching between devices and D for drawing a line using mouse which is no longer used at present.

And, the second group is mental group here everything is related to purely mental activity, no sensory or motor activities are involved. So, in these group there is only one operator M which stands for thinking or decision making task. And, finally, in the system response group there is one operator R which used to refer to the system response time which is no longer valid in the modern day systems.

So, essentially among these seven operators we do not use any more D and R as they are not very relevant nowadays and other five operators can still be used to model interactive tasks in the context of modern day interaction. Now, when you talked of operator we said that they come with a predetermined value. So, their execution time or the predetermined value of these operators were obtained through a extensive studies and those are already reported.

(Refer Slide Time: 13:37)



So, whenever we build a KLM or the keystroke level model for a task we use those values to compute the task execution time as we shall see in subsequent portion of this lecture, but let us see the times that are predetermined for each of these operator values.

(Refer Slide Time: 13:55)

So, as you can see here in this table for K; K operator there are three times mentioned. One is for good typist, one is for poor typist and one is for non-typist; 0.12 second for good typist, 0.28 second for poor typist and 1.20 second for non-typist. These values are obtained through empirical observation empirical studies.

For B or mouse button press or release operator the time is 0.1 second; for P at the pointing task operator the time is 1.1 second, for H the homing tasks switching between devices the time is 0.4 second. As I said before D is not relevant, not important so, we will not bother about their its time. The mental operator M used to take 1.35 second as found out in the empirical studies and R is again not relevant and we will not bother about its time.

So, these are the times that are the values assigned to this seven operators. Among the seven, two operators are not used. So, the values are assigned to the remaining five operator. So, whenever we create a model so, we will use these values to compute the overall execution time. So, we got the idea of operators, we got the idea of their values.

(Refer Slide Time: 16:03)

Now, the thing is we need to understand how to create a model, how to come up with the keystroke level model for an interactive task. So, that is a step by step procedure and, its objective is to analyze or evaluate the design of an interface. So, essentially we will follow a step by step procedure to build a model and use it to analyze or evaluate interface design. So, what are these steps?

(Refer Slide Time: 16:23)

So, broadly we can divide it into three steps. First step is to choose a task, this is important. So, when you are asked to make a model for analysis of a system what you should do is essentially to choose one or more representative task or tasks. Now, what these representative tasks mean? They mean tasks that are likely to be performed frequently by the users. Now, why this is important we have to understand the significance of the use of the word representative. So, when you are analyzing a system, now the system comes with lots of feature. Overall interactive system comes with lots of features which in turn indicate that a user can perform a large number of tasks with the system.

Now, if you are asked to analyze the performance of the system whether the design is good or bad, it is not possible or may not be possible or practical to analyze each and every task because the number of tasks may be very large. So, in that case you have to choose a few tasks from the setup tasks that is supported by the system.

Now, when we are asked to choose few tasks the problem is that if we choose a task which is not likely to be used frequently by the users and if you come up with a conclusion based on the analysis of the task then that conclusion may not be very valid, very significant because it may point out some flaws which in any case is not going to have much effect on the users as they are not likely to perform the tasks or the set of tasks frequently.

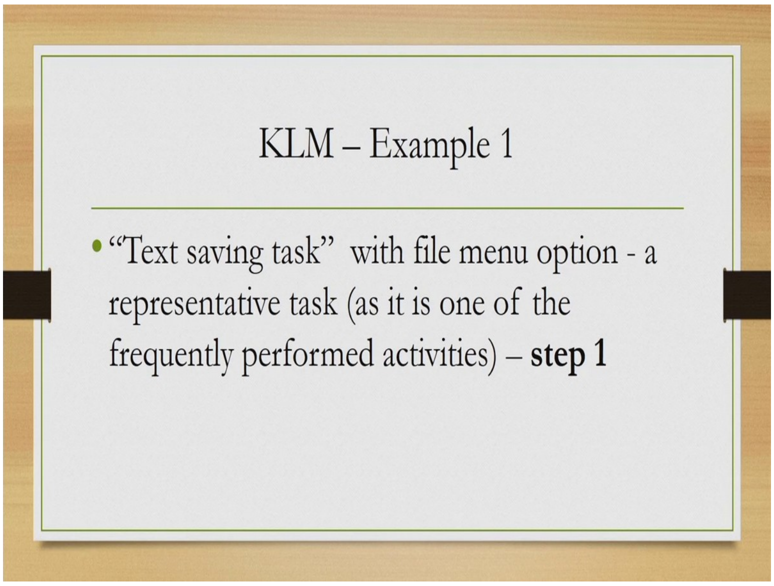
Instead, if you choose one or few tasks that are likely to be performed frequently then any decisions that can be arrived at based on analysis of those tasks are likely to have more significance because most of the users are likely to be affected by the decisions. So, those tasks are representative tasks that are likely to be performed frequently by majority of the users and our first step in building and analyzing KLM is to choose one or more such representative tasks.

Now, once the task or tasks are chosen our next step is to build the model. In this model building exercise the tasks are to identify the operators and arrange them in proper sequence. And, here keep in mind that we will try to identify operators from the set of five operators instead of seven because R and D we have excluded.

So, these five operators are K, P, B, H and M. And, we need to arrange them in proper sequence, that is the second step in the model building and analysis process and in the third stage we need to compute the execution time. So, we need to add up the individual operator times as shown in the previous table.

Now, let us try to understand the idea of building and analyzing a keystroke level model with some examples. So, there are three steps. First step is to identify one representative task, second step is to choose operators and arrange them in sequence and third step is to add up the operator values to get the total execution time. Now, let us go through few examples to understand these model in a better way.

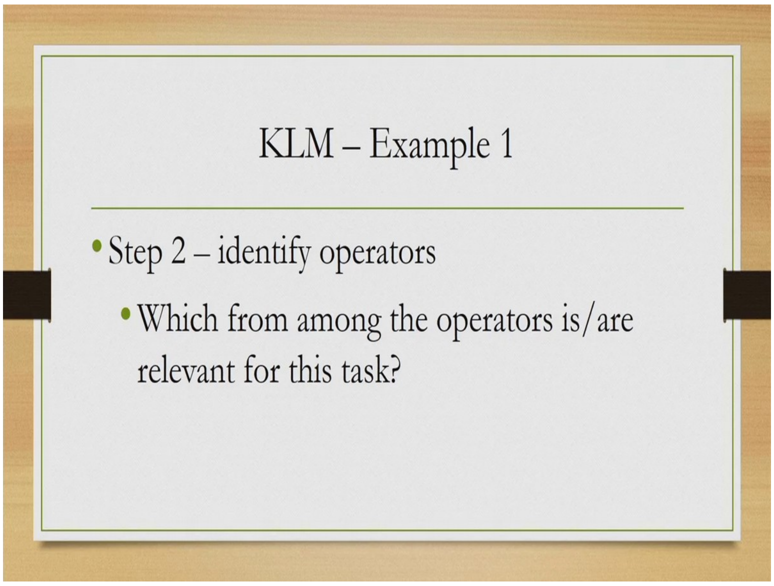
(Refer Slide Time: 20:41)



Our first example is a text saving task. Suppose you are editing a document using a word processor, for the time being let us assume that you are using say MS Word, where there is a menu based save facility available. Now, you are asked to save your editing task whatever you have typed so far, whatever your have edited so far you are required to save it and you are told to do it using the save option.

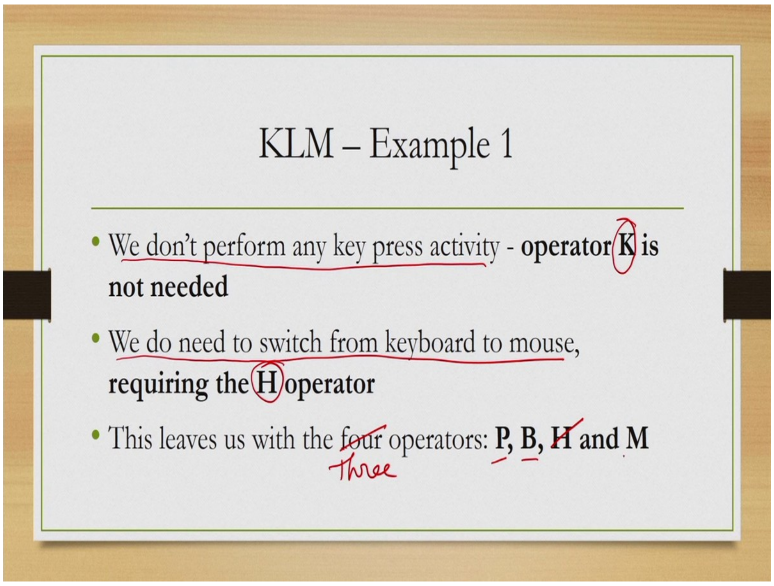
So, what a KLM can tell? KLM can tell us the time it takes to save a file using this save menu option. In order to do that, we need to follow through these steps. Step 1 here is already taken care of we have already said the task is to save a file and we are already assuming that that is representative task.

(Refer Slide Time: 22:07)



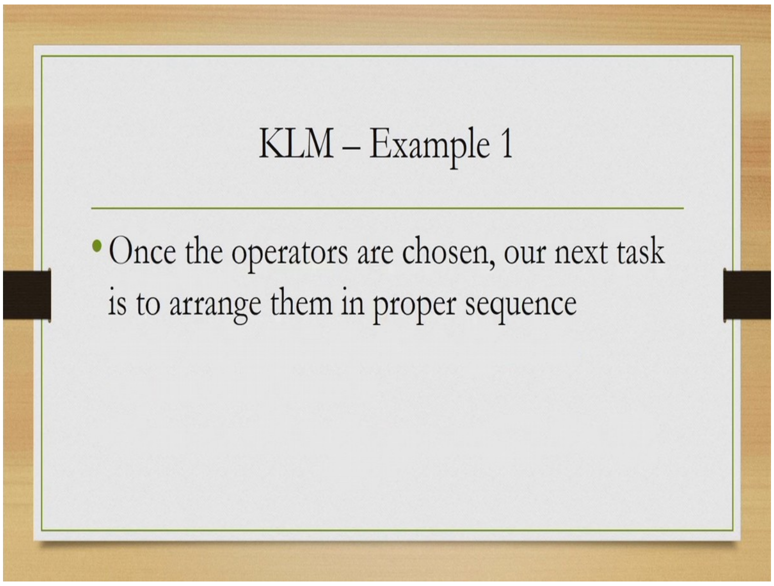
Now, our objective will be to explain step 2 and step 3 that is identify and arrange the sequence of operators and finally, calculate the total time. Now, in this case which from among these five operators namely K, P, B, H and M are going to be relevant to model this task.

(Refer Slide Time: 22:09)



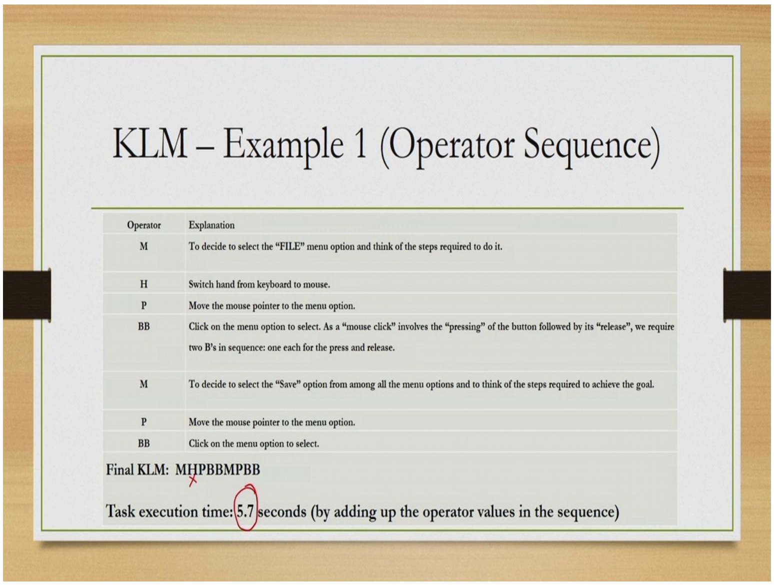
If we analyze the interaction task we may note that we do not perform any key press activity so, K is not required. So, we can exclude the use of K and change it everything is done with mouse. So, we do not need to switch from keyboard to mouse. So, we do not require H operator as well, then this leaves us with only three operators. It should be three namely P, B and M. So, there will be no H operator.

(Refer Slide Time: 22:53)



Now, once the operators are chosen, so, in this case we have chosen P, B and M our next task is to arrange them in sequence in proper sequence.

(Refer Slide Time: 23:03)



So, let us try to do that step by step. So, we start with an M operator, why because we need to decide that we need to select the file menu option and think of the steps required to do it. So, that involves some cognitive activities so, we start with a M which represents our thinking or decision making process.

Now, here all though earlier I said that we do not require H, but here I am assuming that our hand was on the keyboard and we are shifting to mouse. So, we may use an H, but if you are already assuming that our hand is on the mouse then we may ignore it. Then comes P. It essentially refers to the fact that we need to move the mouse pointer to the file menu option.

Then double B. Now, each B represents one press or release. So, when we are required to select the file menu option we need to click on it. Now, your click indicates pressing as well as releasing the mouse button. Now, pressing and releasing together constitute two B operators. So, essentially we need to have two B's one each for press and release. This will be followed by another M. Once I clicked on the or once we clicked on the save option we will get to see another set of menu options and we need to think of selecting the appropriate menu option.

So, once we click on the file menu option, we get to see a set of another menu options among them there is one save option. So, we need to decide to select the save option from among all the menu options that are displayed. So, that is something which we need to do in our mind.

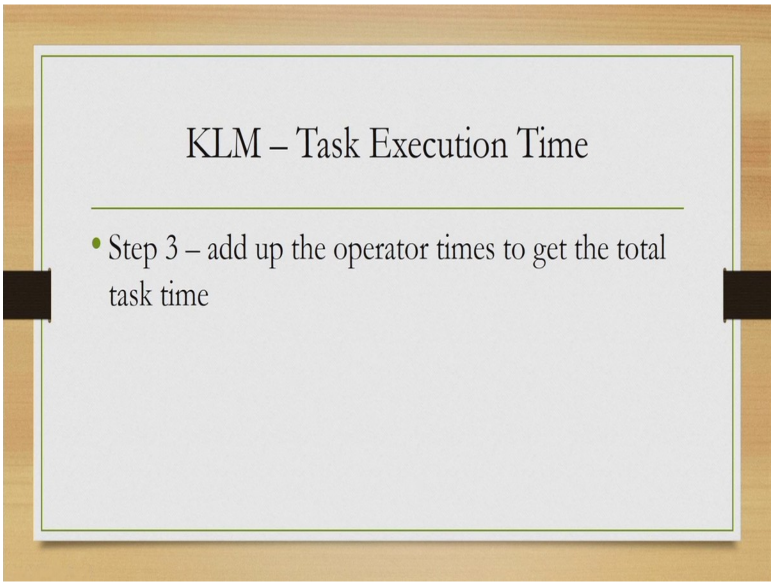
So, that requires another M and once we have decided to select the save option we need to move the mouse pointer to that particular menu option which requires the use of the operator P. This will be followed by another BB to click on the menu option that is save. So, if you recollect BB stands for two operations mouse press and release which together indicate click operation.

So, finally, what we will get is this model of this particular operation which is a sequence of operators MHPBBMPBB. So, to explain it once again we assume that there will be no homing, but here in this operation we are using H which you may ignore if the assumption is that already our hand is on the mouse, otherwise if we are typing and then we are asked to save then probably we need a switch. So, H is required.

So, first we decide to select the file menu option so, that requires one M. This is followed by switching from keyboard to mouse this is one H. This will be followed by moving the mouse pointer to the file menu option that is P followed by a click operation which involves two B operators.

To select the file option this will be followed by a name this M is required to decide to select the save options from among the set of menu option shown after we clicked on the file menu option. This will be followed by a another P which is required to move the mouse pointer to the menu option of save and finally, clicking on the save option or two Bs BB. So, together if we arrange all these operators in sequence we get this model MHPBBMPBB.

(Refer Slide Time: 27:47)

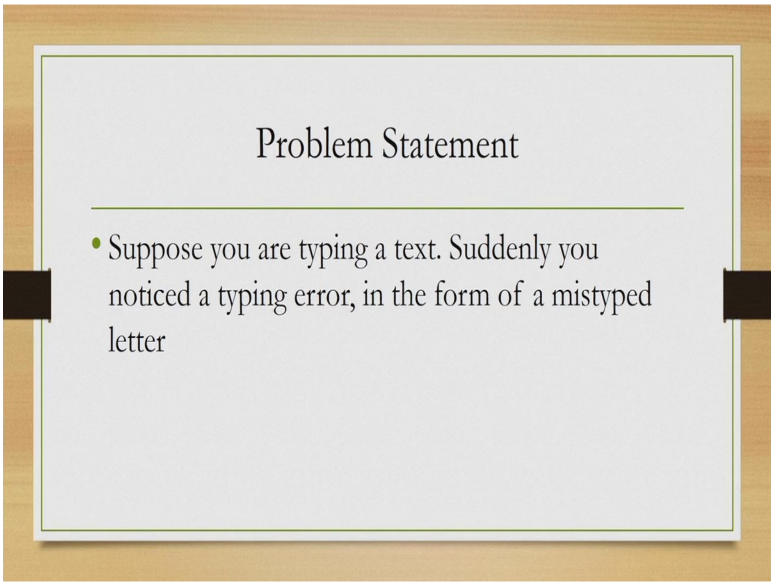


Now, our step 3 is to add up these values to get the total execution time. So, here if we add up the values of M, H, P, B operators then for the total model of saving a file using the file menu option, the total time required will be 5.7 seconds. Now, of course, here we are assuming that our hand was initially on keyboard. Now, if we do not use H then this value will be slightly different other operators will remain the same.

So, two things should be clearer from here when we are building a KLM or Keystroke Level Model the initial assumption is important, based on that assumption we can get different values. So, here if we assume that initially our hand was on keyboard and we need to switch then we need to include H. If we assume that from the very beginning our hand was on mouse that is we have decided and already grabbed the mouse then that H is not required and we can ignore it. So, depending on the initial assumption our values will be different.

And, the second thing is both these values whether including H or not have their significance. So, it is not that if we need to be very accurate at the initial assumption. However, we need to be very cautious as we will see in subsequent examples. So, this previous example was somewhat simple. Let us try to understand KLM with respect to another a little bit more complicated example.

(Refer Slide Time: 29:49)

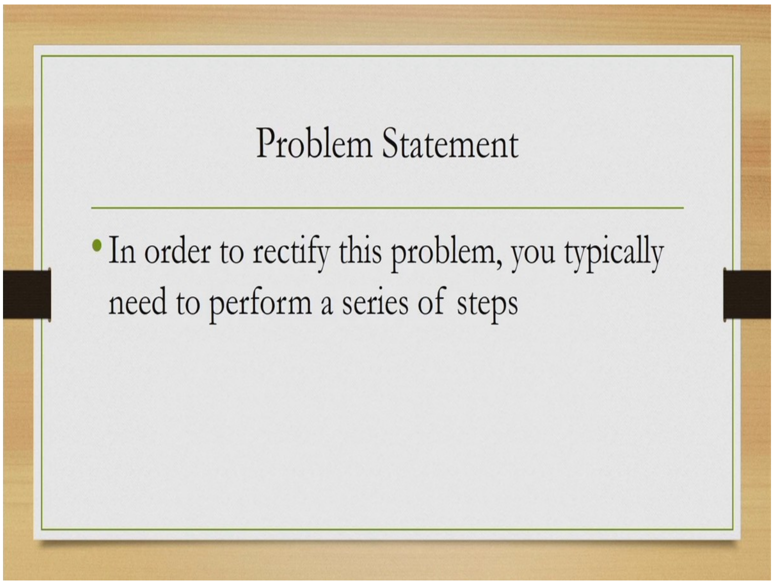


Now, suppose that you want to rectify a typing error. So, you are typing a text using a word editing interface and suddenly you notice that there is an error a character level error and you want to rectify that error. So, in order to do that you need to do certain interaction tasks and our objective is to model those tasks in the form of a KLM or keystroke level model.

(Refer Slide Time: 30:17)

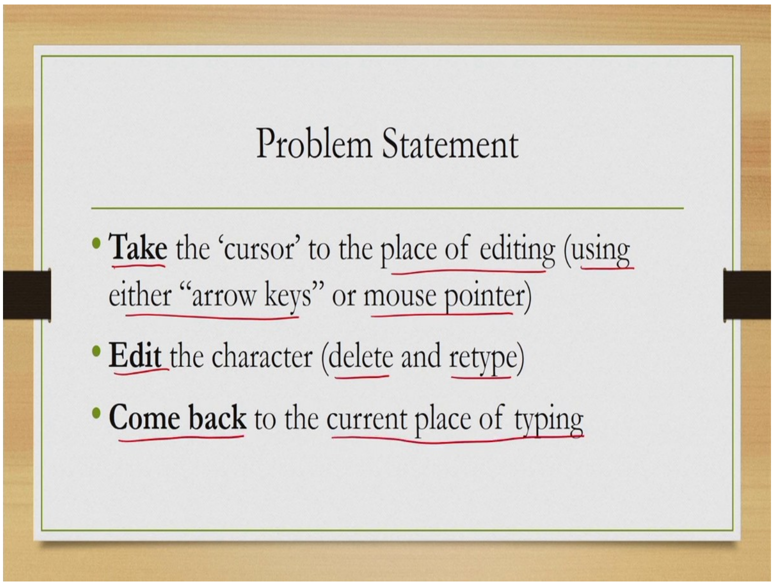
So, what we need to do? Let us explain the problem situation with this figure. Here as you can see there is a text on the interface you are typing currently you are at the end of the text here, but then at this point you have notice that you have made a typing error here at this point and you want to rectify it. So, then what are the things that you need to do and how to model that using KLM that is what these example is all about.

(Refer Slide Time: 30:51)



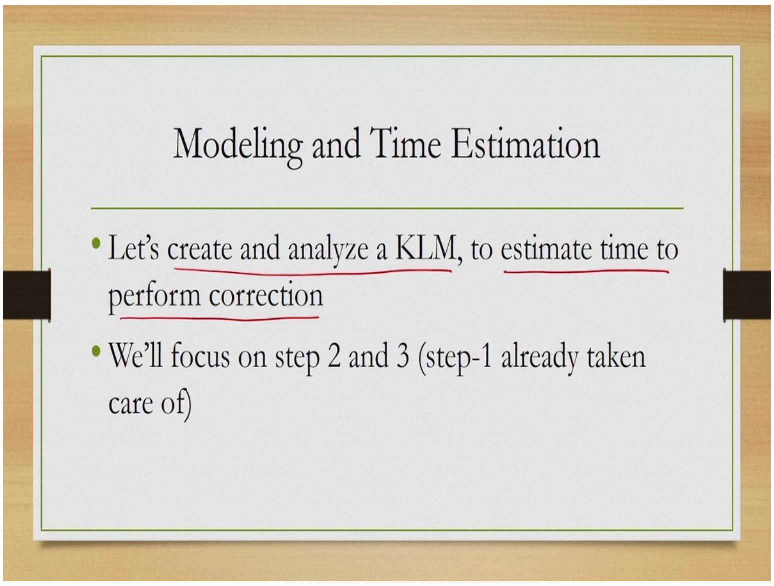
So, in order to do this you need to perform a series of steps. What are those steps?

(Refer Slide Time: 30:55)



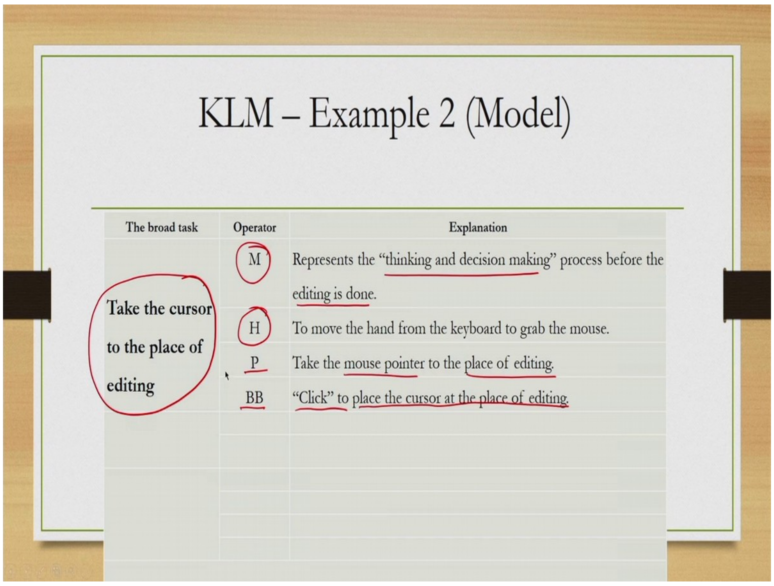
You need to take the cursor to the place of editing you may use either arrow keys or mouse pointer then you need to edit the character; that means, deleting the wrong character and retyping the correct character and then come back to the point from where you shifted to the point of error.

(Refer Slide Time: 31:25)



Now, for this let us try to create and analyze a KLM. These KLM will tell us how much time it will take to perform the correction and again already it step one that is choosing a representative task that is already taken care of our task is to correct an error. So, we will focus on step 2 and 3.

(Refer Slide Time: 31:49)

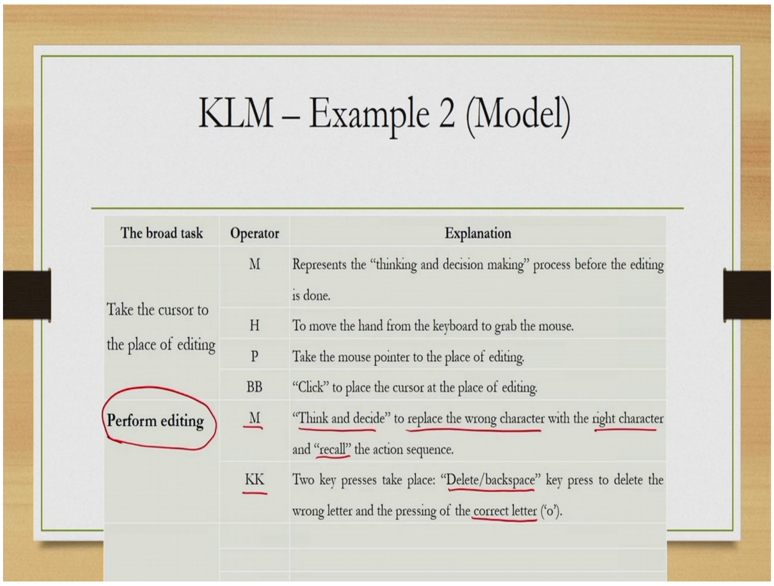


So, let us see the first task that is take the cursor to the place of editing. So, here we will assume that we are using a mouse pointer to take the cursor rather than any keys arrow keys. So, then first thing is mental operator M which represents the thinking and decision making process before the editing is done. So, we are making a decision that editing is required and we need to do it so, we have to do certain steps. So, that decision is represented by M.

Now, remember that your typing. So, then your hand was supposed to be on the keyboard and in order to use your mouse to place the cursor at the right place you need to make a switch from your keyboard to the mouse and that is taken care of by an H operator.

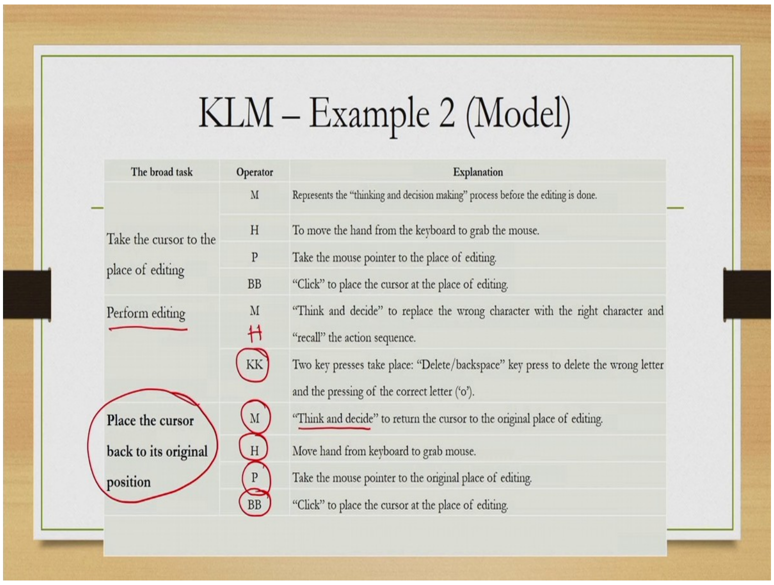
This is followed by P which essentially refers to taking the mouse pointer to the place of editing and this is followed by BB that is Click the mouse, so that the cursor is placed at the right position. So, then for the first task that is taking the cursor to the place of editing we require the operators in this sequence M, H, P, BB.

(Refer Slide Time: 33:13)



Then, the next task is to perform the editing actual editing. So, here we require an M operator to think and decide about replacing the wrong character with the right character and recall the action sequence how to do that. Then this is followed by two case; one K is for deleting the wrong character and the second K is for inputting the correct character. So, then the perform editing task can be represented by the operator sequence M, KK.

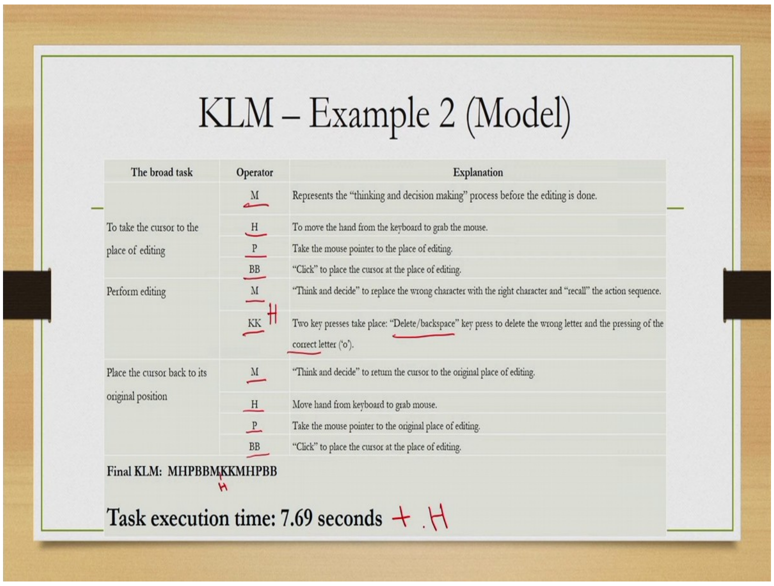
(Refer Slide Time: 33:55)



Then comes the third and final task that is placing the cursor back to it is original position. So, here we need a think operation represented by an M, then homing operation represented by H, mouse pointing operation represented by P and another click operation represented by BB.

Now, in this sequence do you think we missed something? So, here as you can see we used one H to move hand from the keyboard to grab mouse; that means, somewhere we grab the keyboard that was here when we press the delete or backspace key and the correct letter, but then before that our hand was on the mouse when we clicked to place the cursor at the editing.

So, there has to be another H somewhere which should have been at this stage, perform editing which we did not use. So, here along with M there should be one H followed by KK. (Refer Slide Time: 35:09)



So, then our entire sequence will be something like this MHPBBMKK where after M there should be one H, then MHPBB. Now, this sequence represents the cognitive activities involved in performing the interaction task of correcting a mistyped character during text editing. And, from there we can calculate the total time which is 7.69 seconds excluding this H. So, the total time should be 7.69 second plus H.

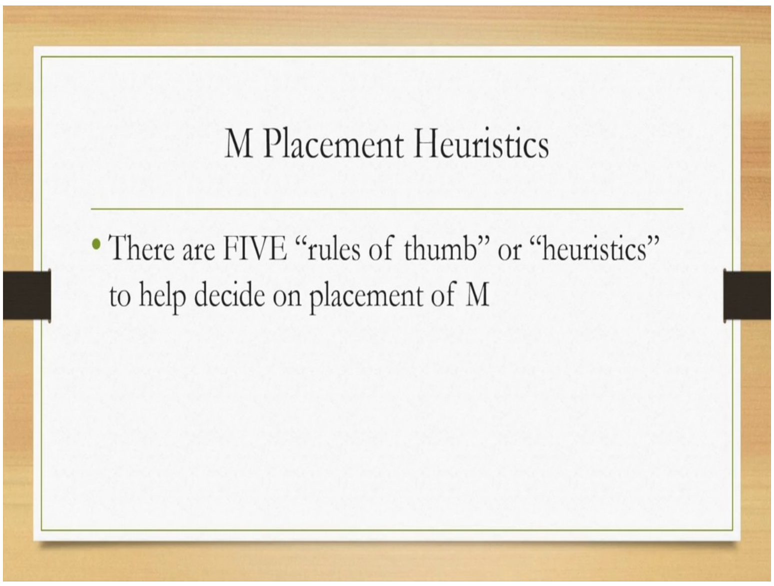
So, in order to understand it let us go through it again. So, our task was to rectify a mistyped character. So, we assume that we are at the end of typing at a particular location and we noticed a mistyped character somewhere and we want to rectify it. So, then first we need to decide that we need to rectify it. So, that is one M. This is followed by one homing operator because we are typing so, we need to shift to mouse. This is followed by a P which is basically placing the mouse pointer at the right place.

Then clicking the mouse pointer so that it is placed at the right location that is a collection of operators BB this is followed by the actual editing activities which has one M for initial decision making, then one H for homing again, then one KK or the series of two operators K and K representing the two operations delete and insertion of the correct letter.

And finally, we want to come back to the point from where we left. So, that is again making a decision at the beginning so, that is M followed by H homing operator. So, from keyboard we want to get back to mouse then P. So, bringing the mouse pointer to the correct position and then placing it there by click or BB. So, the sequence is MHPBBMHKKMHPBB and the total time is 7.69 second plus time for homing which is not shown in this sequence.

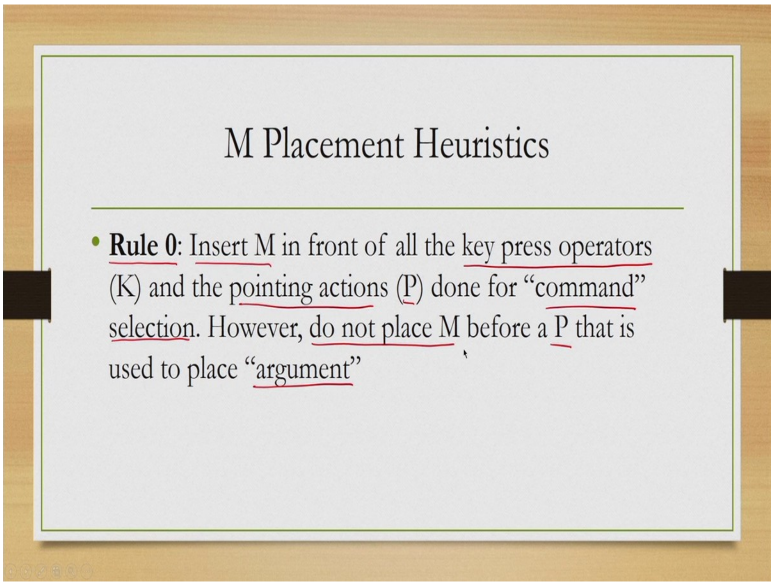
Now, from these two examples you probably have got some idea of how to construct keystroke level model. It is relatively easy as you can see from the examples only five operators out of which you have to choose the right operators and place them in sequence adapt their values to calculate the total time. Placing of most of the operators is not very difficult except the mental operator M. Sometimes it is very confusing, it is not very clear where to place M and we sometimes get confused about its right placement.

(Refer Slide Time: 38:43)



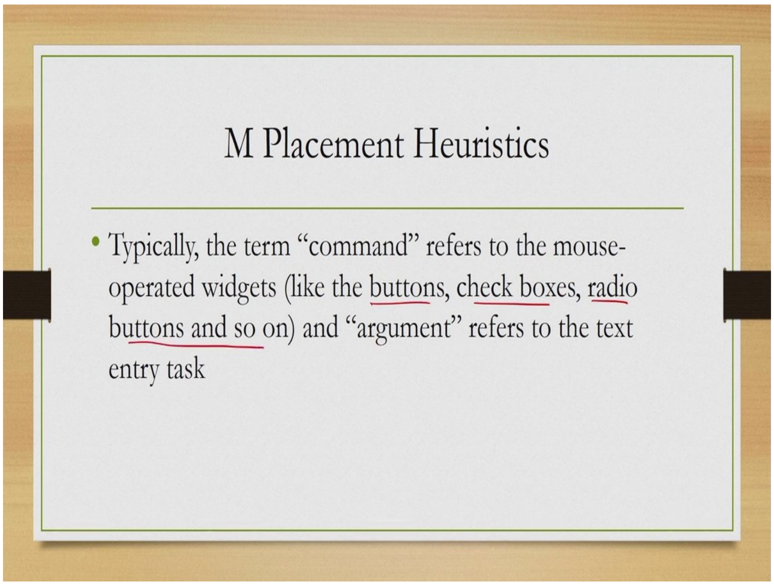
So, in order to assist you to build a model with M there are few heuristics given. Although there are six heuristics we will talk about five. One is no longer used we will mention it at the end and the heuristics are essentially thumb rules rule of thumbs which tells you where to place M and based on what logic. So, let us go through these five heuristics one by one. These are as I said again rules of thumb for the placement of M.

(Refer Slide Time: 39:25)



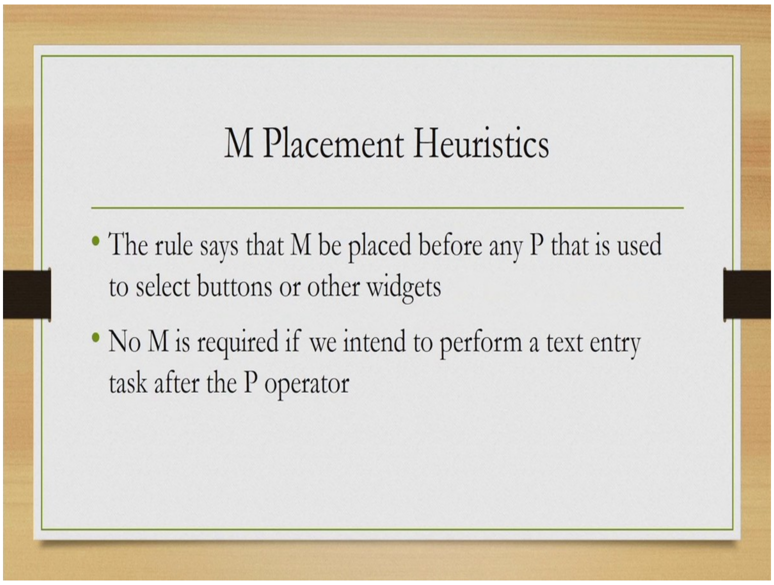
So, the first rules tells us that insert M in front of all the key press operations; that means, wherever we are having a K and pointing actions wherever we are having P, but not for all P pointing actions that are done for command selection. Do not place M before a pointing action that is used to place argument. So, place M before all key press operators and place M before some P operators, but not all P operators. So, what are these things? Command and argument.

(Refer Slide Time: 40:13)



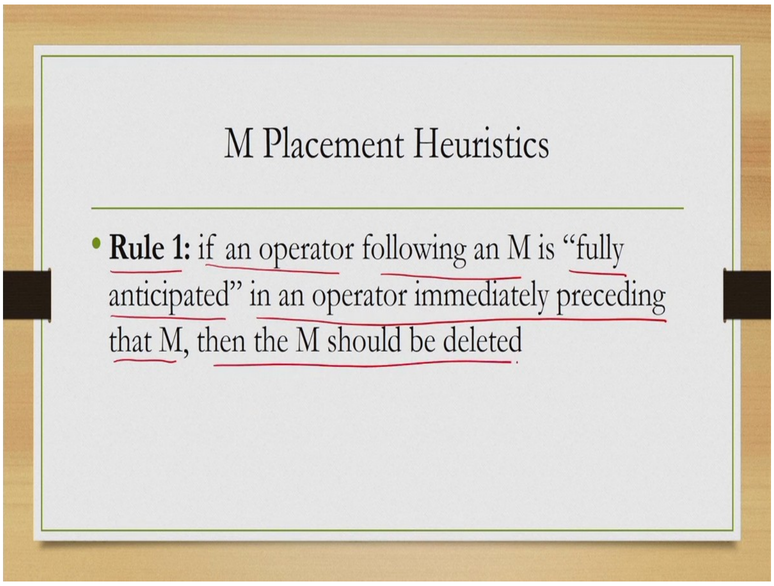
So, when we talk of command typically what we refer to is some mouse operated widgets like buttons, checkboxes, radio buttons and so on and argument refers to text entry tasks.

(Refer Slide Time: 40:27)



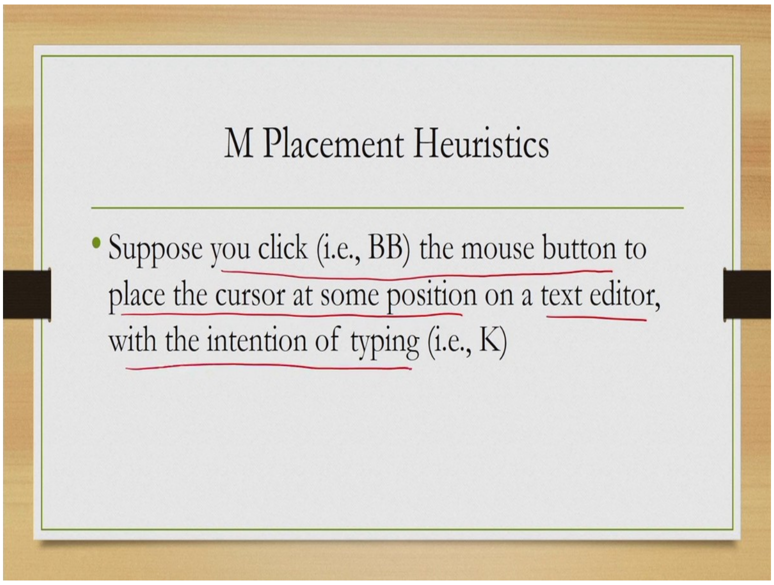
So, in other words what these rule 0 tells us is that whenever we are using the mouse to select button or checkbox or some other similar widgets. So, before all those Ps we should place M. However, if we are using the mouse to place cursor for typing which is typically considered as argument then we do not need to place any M before that.

(Refer Slide Time: 40:55)



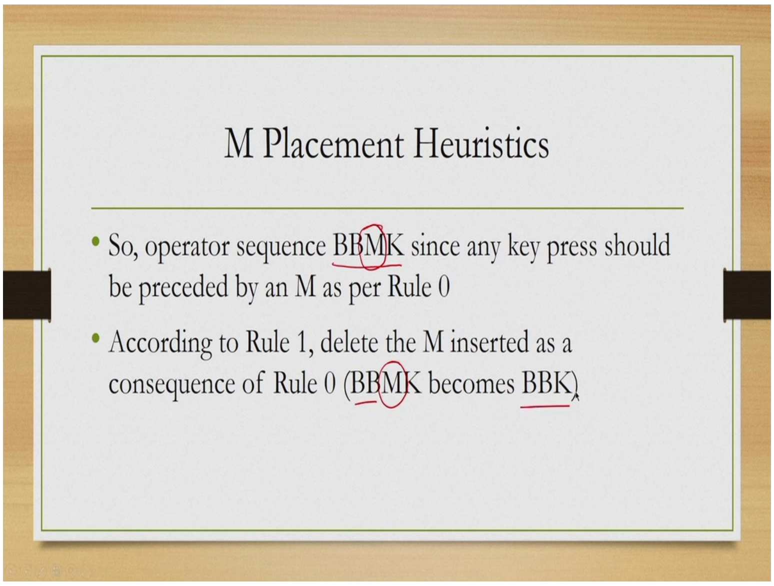
Then comes rule 1. So, earlier one was rule 0, second heuristic is rule 1. It says that if an operator following an M is fully anticipated in an operator immediately preceding that M, then the M should be deleted as an example.

(Refer Slide Time: 41:17)



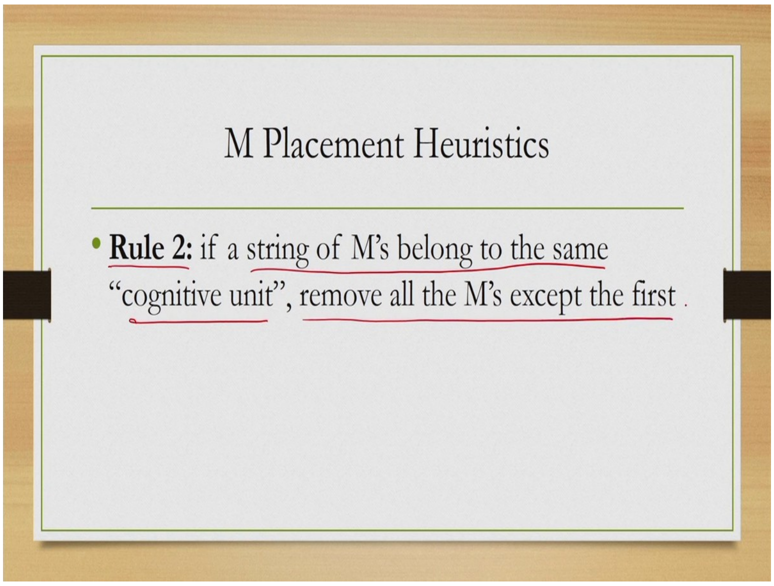
Suppose, you click the mouse button using of course, the operator sequence BB to place the cursor at some position on a text editor with the intention of typing K.

(Refer Slide Time: 41:33)



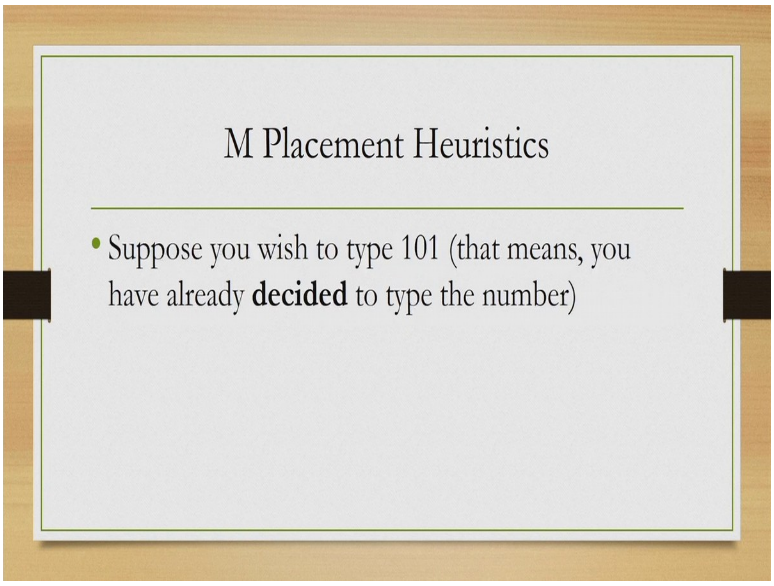
I mean in that case as per rule 0 it should be BBMK that should be the correct operator sequence. According to rule 1 this M may be removed because it is already anticipated in BB. So, whenever you are placing the cursor you already are aware of it that you are going to type it. So, it is already anticipated in BB that there will be an there will be a K. So, this M is no longer required. So, the BBMK becomes BBK. So, that is the idea of rule 1.

(Refer Slide Time: 42:15)



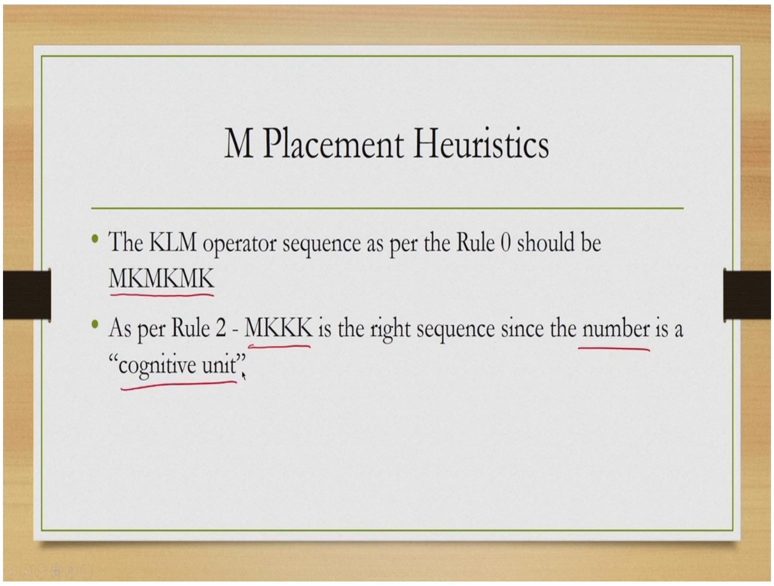
Now, rule 2 says that if a string of Ms belong to the same cognitive unit then remove all the Ms except the first. Let us see another example.

(Refer Slide Time: 42:27)



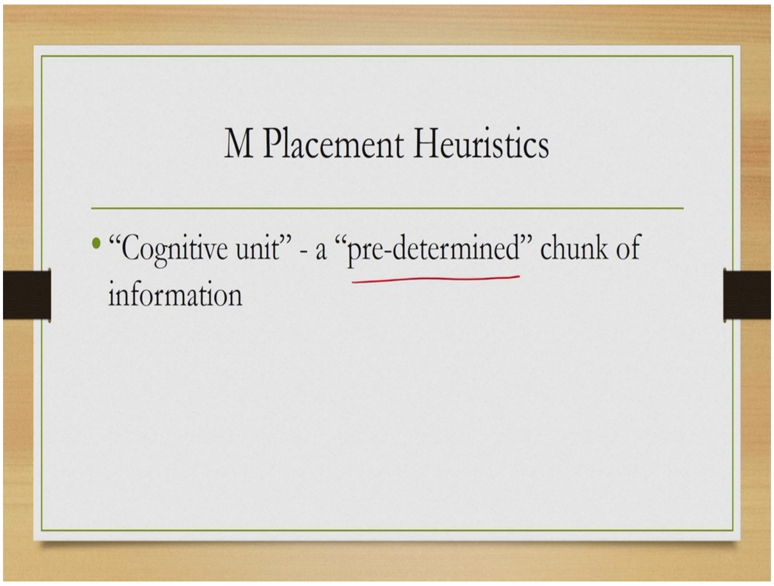
Suppose, we want to type 101, this is a number.

(Refer Slide Time: 42:39)



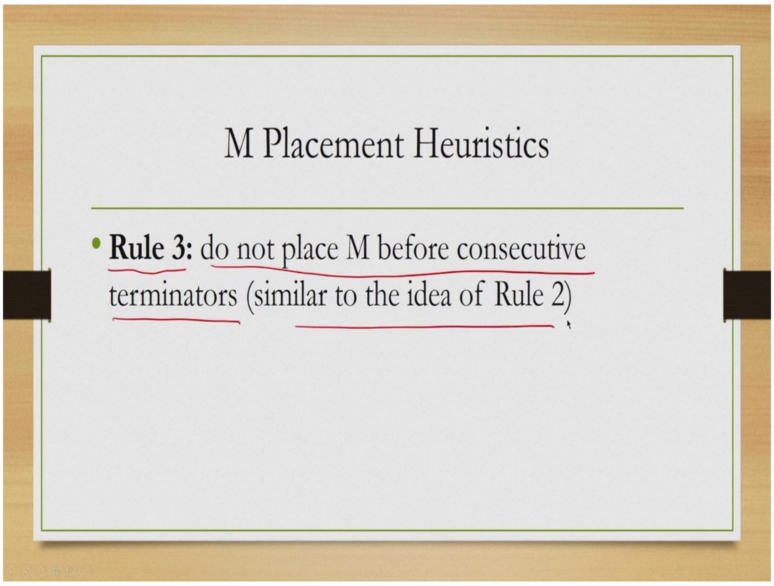
Now, as per rule 0 since there are three K operators, so, ideally your operator sequence should be MKMKMK, one M before each K. But, as per rule 2, it should be MKKK because the number is a cognitive unit.

(Refer Slide Time: 42:59)



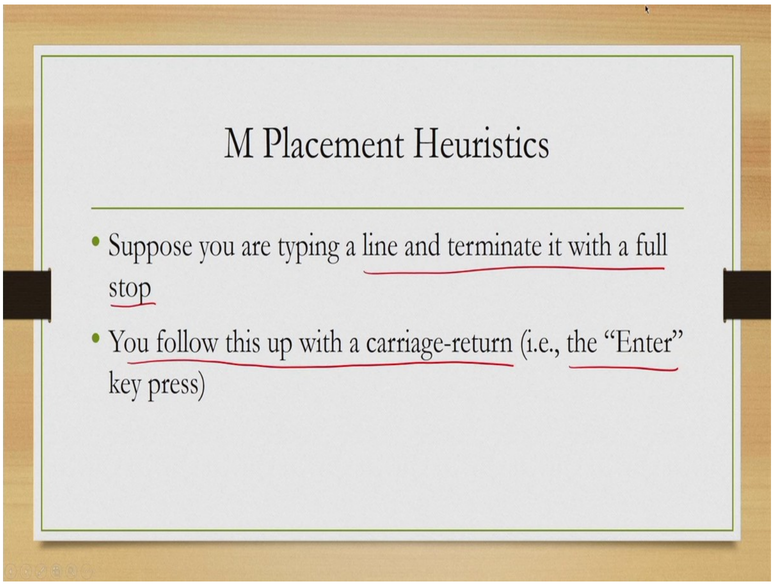
So, what is a cognitive unit then? It is a chunk of information that is already predetermined. So, if you have a predetermined chunk of information that you are going to type, then you require only one M instead of one M before each K when you are typing. And, the logic is simple. Since you have already decided to type these entire chunk of information before hand so, only one M should be placed for taking that decision.

(Refer Slide Time: 43:27)



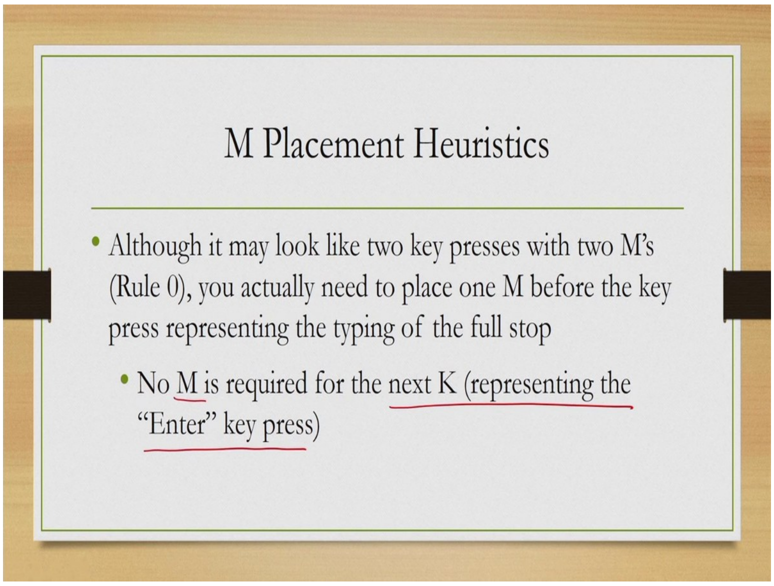
Then, comes the third rule the forth heuristic or rule 3. It says that do not place M before consecutive terminators. In a way it is similar to the idea of rule 2.

(Refer Slide Time: 43:43)



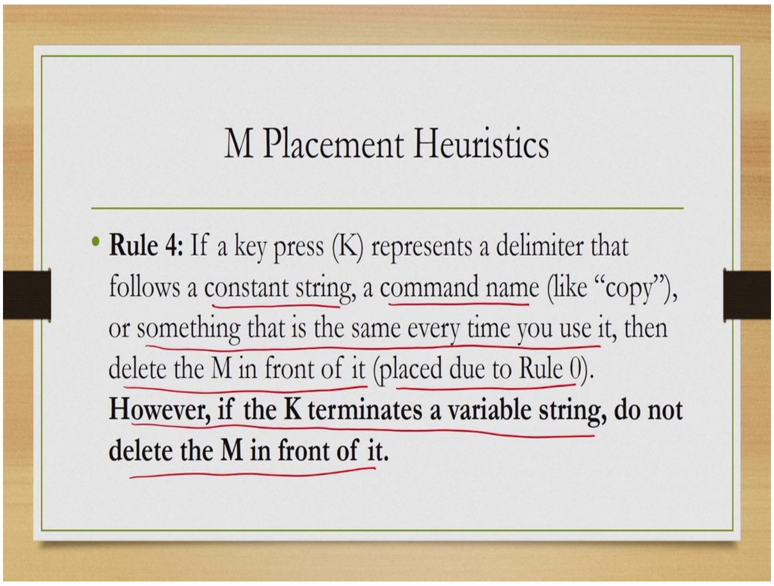
As an example suppose you are typing a line and terminate it with a full stop. So, at the end of the line you placed a full stop then you want to go to the next line. So, you follow it up with a carriage-return or press the entire key. So, here there are two terminators, a full stop followed by a carriage return.

(Refer Slide Time: 44:07)



Now, in this case ideally as per rule 0, you should have one M placed before each of the key press, one for full stop and one for carriage return, but according to rule 3 we do not need to place any M for carriage return. So, only one M is sufficient for both terminators.

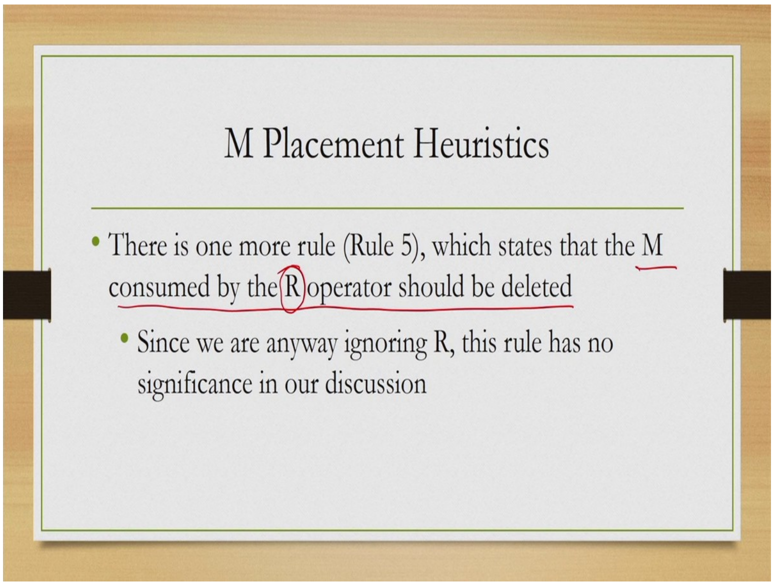
(Refer Slide Time: 44:39)



And, the final rule is rule 4, this is the fifth heuristic it says that if a key press represents a delimiter that follows a constant string a command name or something that is the same every time you use it, then delete the M in front of it which is placed due to rule 0. However, if the K terminates a variable string do not delete the M in front of it.

So, if your typing the same command every time and ending it with a delimiter key then as per rule 0 before that delimiter key also you need to place an M. However, that is not required as per rule 4 if the string that you are typing remains the same always, but if the string changes every time you type then you need to place that M that is what rule 4 or the fifth heuristic tells us.

(Refer Slide Time: 45:45)



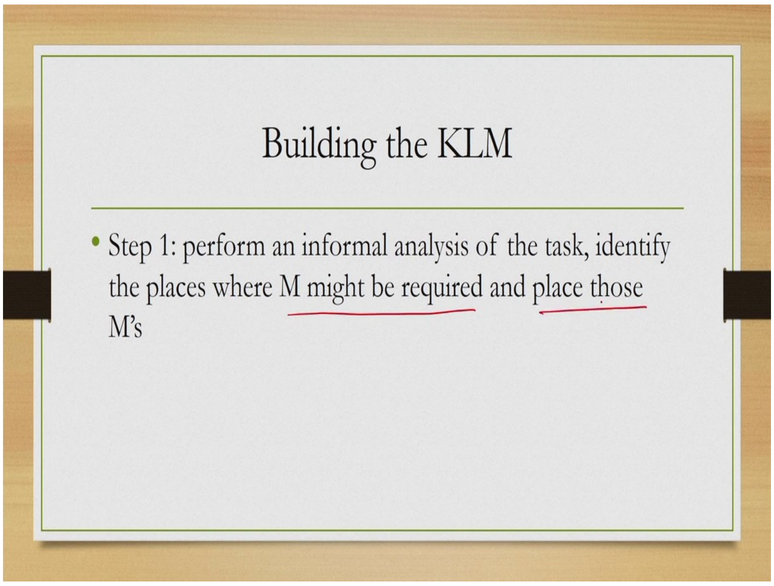
And, as I said apart from this five heuristics there is one more which was developed to take care of the situation that was prevailing at the time of 1980s, but it is no longer valid now. It says that M consumed by R operator should be deleted. So, remember R is the system response time. So, now, since the response time is not significant. So, this anyway does not hold any relevance for modern day interactive tasks.

So, then if we summarize what we have learnt so far is that in KLM we are assuming that there are operators which are low level cognitive tasks. These operators are placed in sequence to model and interactive task and there are stages which we need to follow to model the task.

These are first choose a representative task then identify the operators that are required to modulate and place those operators in proper sequence and finally, add up the values of these operators to get the task execution time. And, as the examples show these task models are essentially models of our thinking and sensory motor activities. So, essentially they represent cognitive and motor activities of the human user.

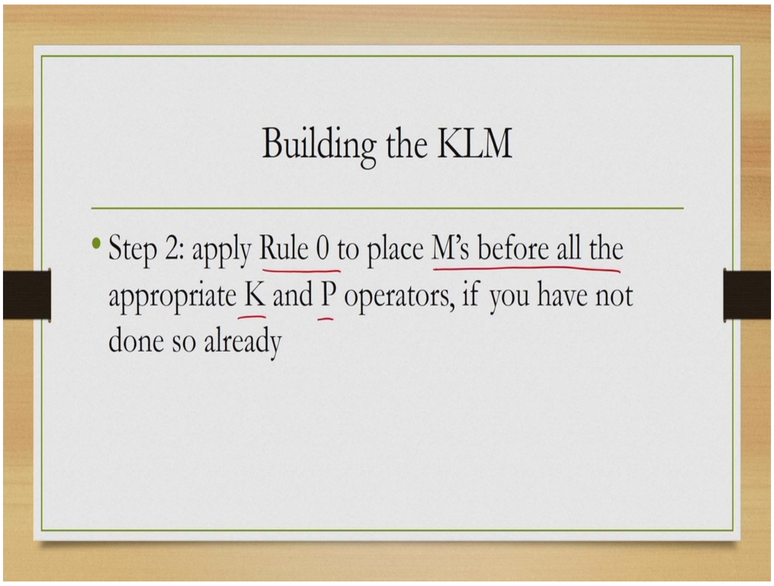
So, when we talk of interactive task we talk of the tasks that we perform mentally or through our motor organs or sensory organs and each of these operator has predetermined value which have been obtained through extensive empirical studies in the past.

(Refer Slide Time: 47:43)



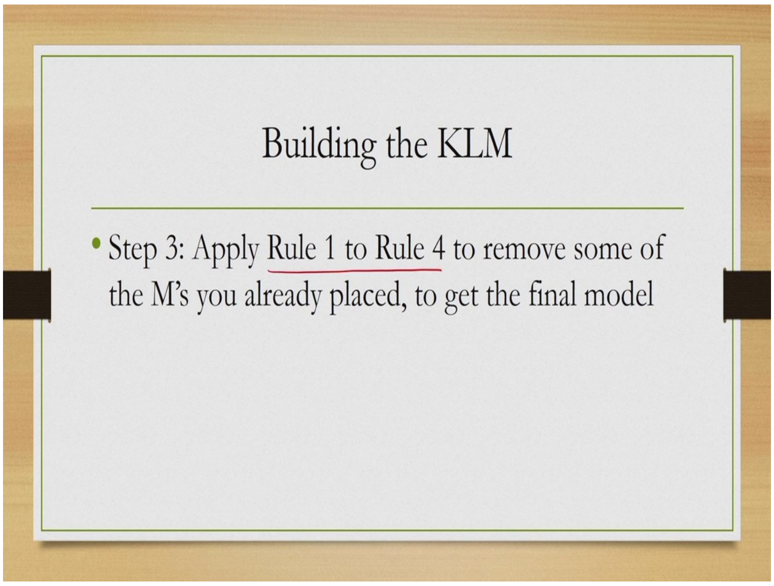
So, in order to build a KLM or keystroke level model, then what we need to do? In the step 1, we need to perform an informal analysis of the task and identify the places where M might be required and place those M's. So, when we want to build a model we start informally as we have done in the examples and there we place them as per our intuition.

(Refer Slide Time: 48:13)



This is followed by application of rule 0 which tells as where to place M's so, we put M before all the case and appropriate P's. Now, some of these M's might already have been there because our inform analysis some may not be. So, we apply rule 0 to complete this.

(Refer Slide Time: 48:35)

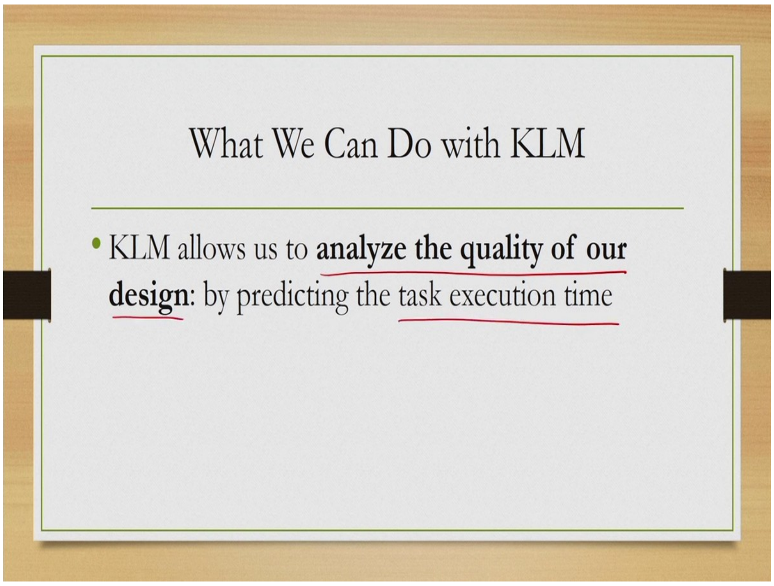


And, then in step 3 we apply the other rules rule 1 to 4 to remove some of the M's that are needed to be removed as per the rules and after application of these rules what we get the sequence that we get is the model, final model.

(Refer Slide Time: 48:59)

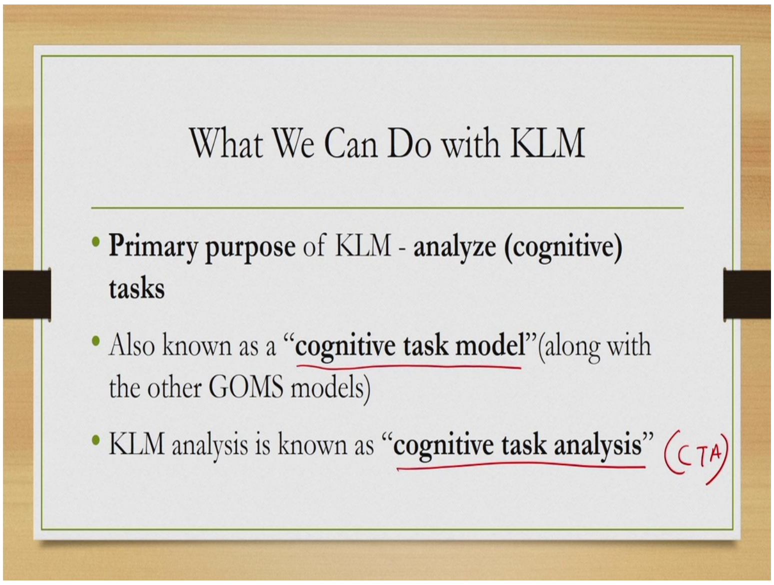
So, then how we can use KLM? So, already we said that we need to build KLM to compute or predict the task execution time.

(Refer Slide Time: 49:07)



So, what is the use of this time? So, with this time we can actually analyze the quality of the design. So, this is an important use of KLM we can analyze the quality of our design with the task execution time that we have predicted using KLM.

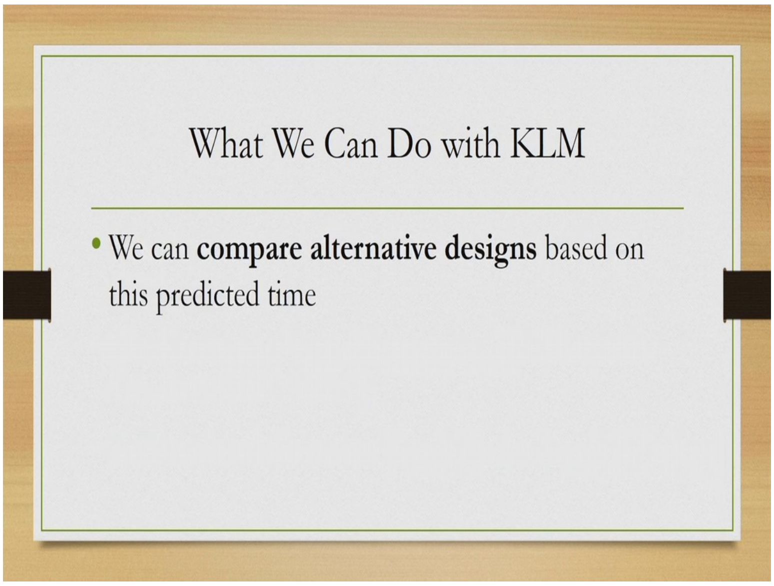
(Refer Slide Time: 49:21)



So, when we want to analyze the quality of the design the analysis is done in terms of the cognitive tasks that take place in the mind of the user as we have already made clear. So, essentially what we are doing is analyzing the tasks from a cognitive point of view. So, this type of analysis is also known as cognitive task analysis and the corresponding models that are used to perform these analysis are sometime known as cognitive test models.

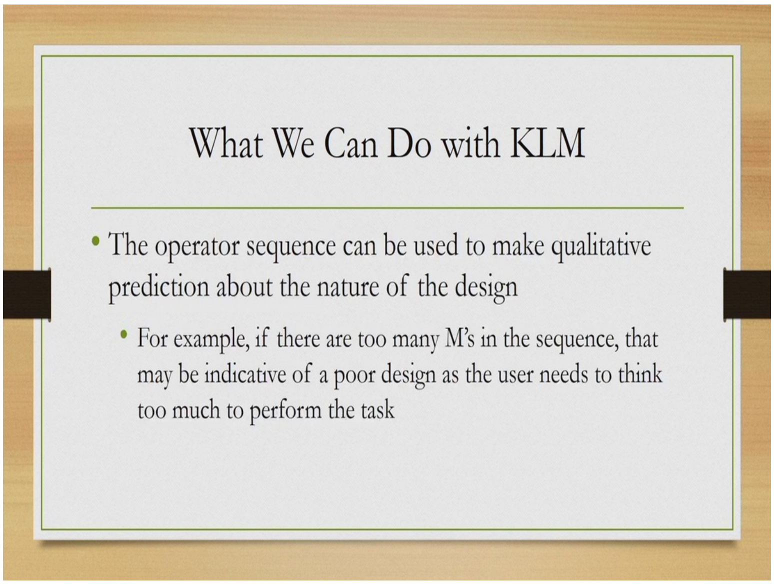
KLM is one example of such models and KLM analysis is one example of such cognitive task analysis or CTA. There are other models other GOMS models fall into this category plus there are other models which we will not have time to discuss in this course.

(Refer Slide Time: 50:17)



So, how do we analyze the quality of our design? By comparing designs alternative designs.

(Refer Slide Time: 50:27)

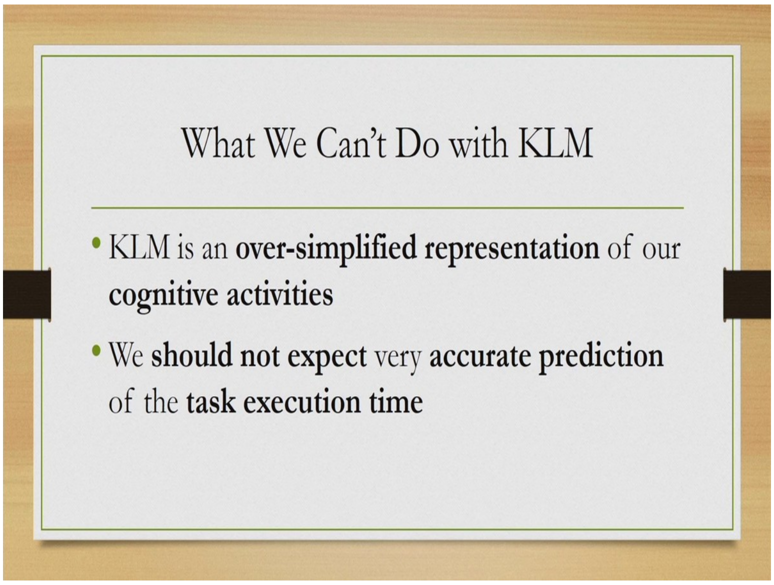


So, suppose there is design 1 and there is design 2 and in both the designs say for example, a text entry interface where we have certain ways to save a file. Now, in design one we compute the time to save a file and in design 2 we compute the time to save a file. Then we compared these two times. So, the time that is less points to a design which is likely to take less time of the user to perform a frequently used task which indirectly indicates that the design is likely to be more efficient which is one important measure of usability.

Now, that is one way of using KLM. There is another way. The sequence can also be an indication of the quality of design. The longer the sequence is to perform a frequently used task we can say that the design may not be very usable because the user need to perform lot of cognitive tasks to perform a interactive task instead if the same task can be performed with a shorter sequence of operators which indicates sorter list of cognitive activities.

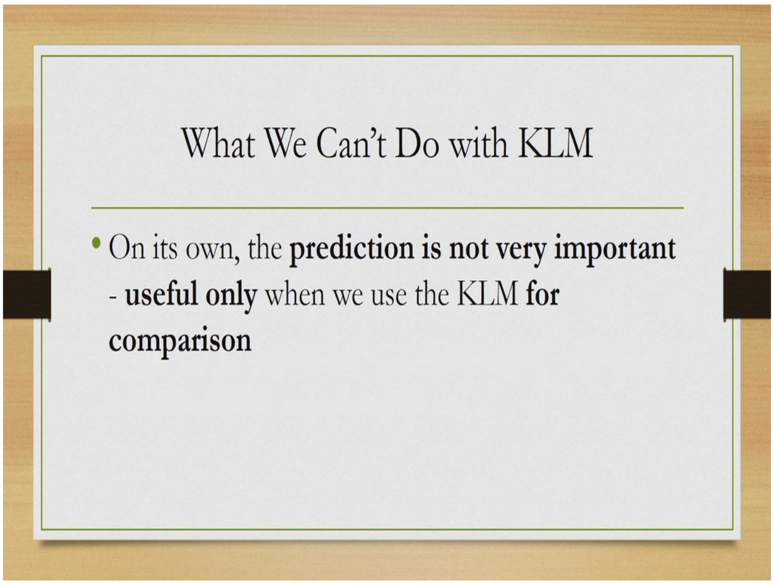
Then of course, indirectly that design indicates more usable design because it puts less pressure on the user's cognitive process. So, that is qualitative way of using the operator sequence. Quantitative way is through the use of task completion time we compute the times and then compare alternative designs; qualitative way is through the use of the length of the sequence.

(Refer Slide Time: 52:19)



However, it must be kept in mind that in KLM we have made very simplified assumption about human thinking that is any cognition can be represented as a sequence of low level operators, a linear sequence and there is no consideration of parallel thinking. So, this may not be very realistic, but it has its practical value.

(Refer Slide Time: 52:43)

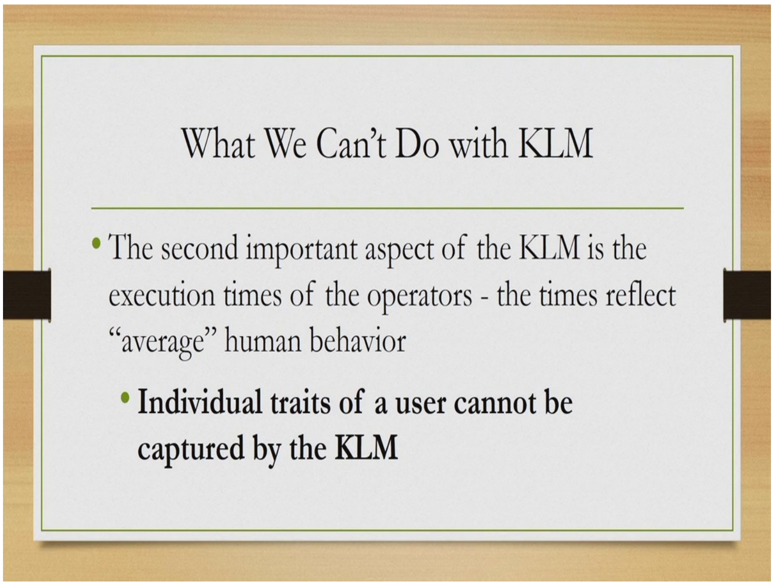


And, we should not try to make out much from the actual value. The actual value does not signify much because we have made very simplistic assumptions. However, when we compare two designs or compare multiple designs then the value make some sense in a relative way.

It tells us in a relative way which one is better, but absolute value of task execution time has no great significance that you should always keep in mind. There is another problem with the KLM. So, when we talk of operator times as we said these times were determined based on extensive user studies.

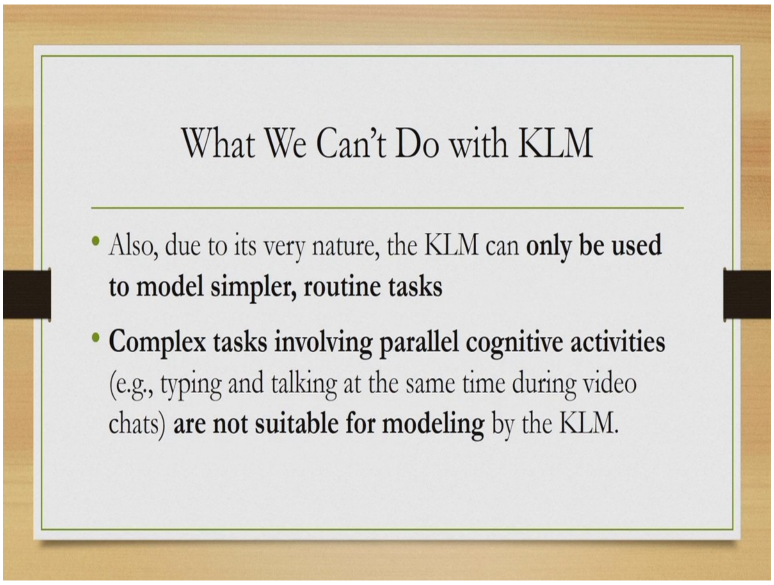
Now, these studies were carried out with large number of users and the values that we ultimately use is an average of the values that were obtained from all the users. So, essentially what it means is that the values operator times that we use are representative of average user behavior rather than individual behavior. So, individual values may vary from the values that are used.

(Refer Slide Time: 53:57)



So, it does not or it cannot take care of individual traits of a user such as the time it takes for an individual to actually type or point or click, these things cannot be captured instead what we are modeling is behavior of average user.

(Refer Slide Time: 54:15)



And, finally, since we are assuming that cognition is a very simple process without any parallel activities involved. So, what we can at most expect is that we can model very simple routine tasks with KLM with the language, with the specification that KLM supports. Complex parallel cognitive activities that are required in most of the modern day interactive tasks may not be completely modeled with KLM.

So, to recap we have planned about one of the models in the GOMS family that is KLM or keystroke level model. This is amongst the earliest predictive engineering models that are used for user-centric system design. It was first proposed in 1980. So, it consists of seven operators that are arranged in sequence to build a model. Among these seven five are relevant today, two are no longer relevant. So, these five operators are the K operator, the P operator, the B operator, the H operator and the M operator.

So, using these five operators we can build models for interactive tasks where these models essentially referred to models of our cognitive activities during performing those tasks and when we build a model we can use the model to compute the task completion time by using the predetermined values for these operators. So, the model is nothing, but a sequence of these operators and we add up the individual operator values in the sequence to get the total task execution time.

Building the model is not very difficult, but sometimes placing of the M operator create some confusion, create some difficulty. In order to address that there are five heuristics or rules of thumb proposed using those five heuristics we can build the model with much ease.

And, the model can be put to use in two ways: one is by comparing the task execution time of alternative designs we can conclude about the quality of the design. The less time it takes to perform the same task the design is better also by looking at the length of the sequence of operators to perform an interactive task we can come up with some qualitative judgment about the quality of the design, the longer the sequences the design may not be very good. We need to reduce the sequence otherwise the cognitive pressure on the user will be more that is a qualitative way of looking at the sequence.

However, KLM is a very simplistic representation of our cognition and we should not try to make much out of the task execution time the value that is computed using the model. Only in a relative sense during comparison we can use it and using KLM we can model simple routine tasks rather than complex tasks which involve parallel cognition.

(Refer Slide Time: 57:39)

The topics that we discussed today can be found in this book. You are advised to look at Chapter 4, Section 4.2.2 to get more details including the relevant references on this topics that I covered in today's lecture.

Thank you and goodbye.