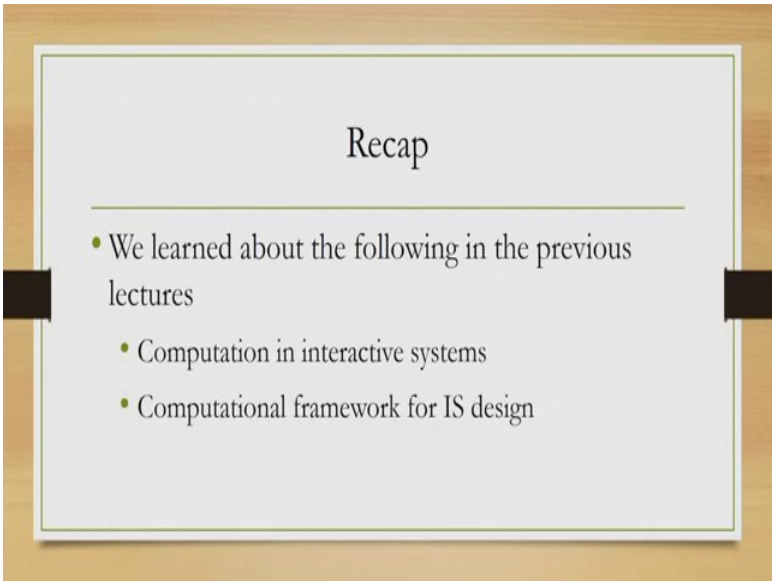
User-Centric Computing for Human-Computer Interaction
Prof. Samit Bhattacharya
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture - 11
User centric models – introduction and descriptive models

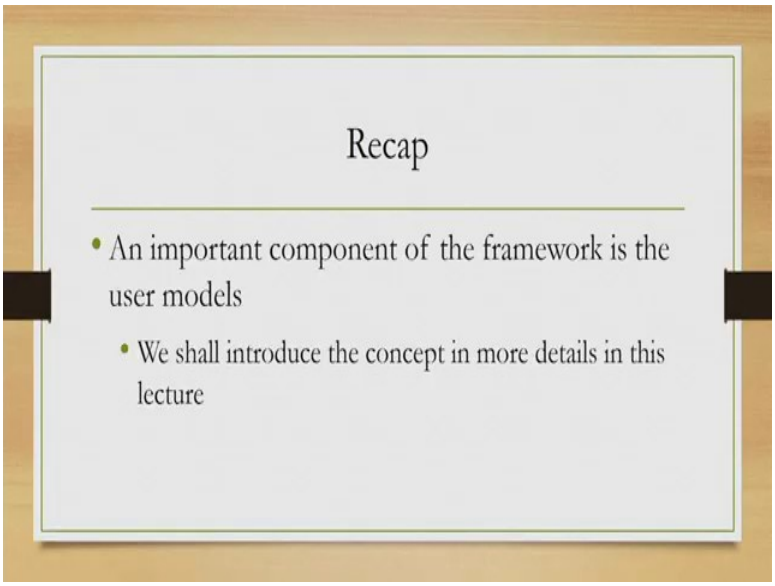
Hello and welcome to the 11th lecture of the course User Centric Computing for Human Computer Interaction. So, as usual before we move forward I would like to briefly recollect what we have learned so far in our earlier lectures.

(Refer Slide Time: 00:54)



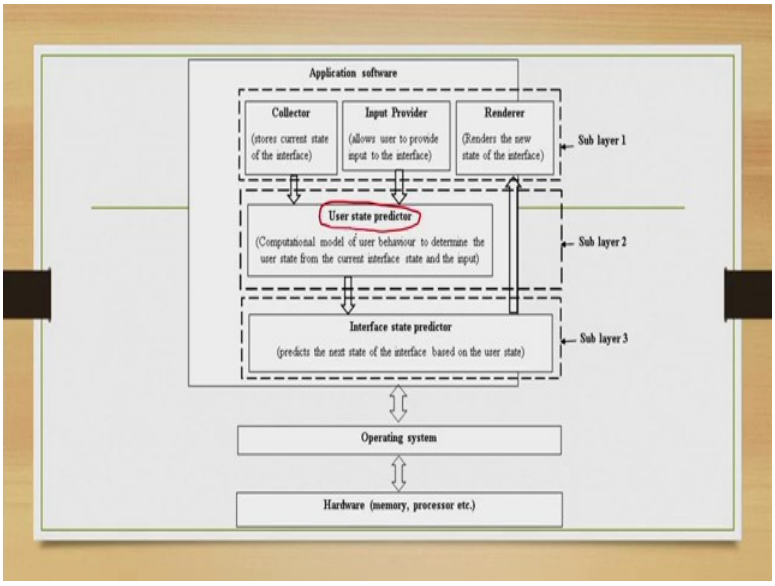
So, far we have learned about two things: one is the idea of computation in the context of interactive system software and the idea of a framework that helps us in understanding this computational aspect of user centric systems.

(Refer Slide Time: 01:20)



Now, if you may recollect there is one important component in the framework that made it different from the other standard software development process that is the user models. So, where this user models fits in the framework let us try to recollect with the help of the framework illustration.

(Refer Slide Time: 01:43)



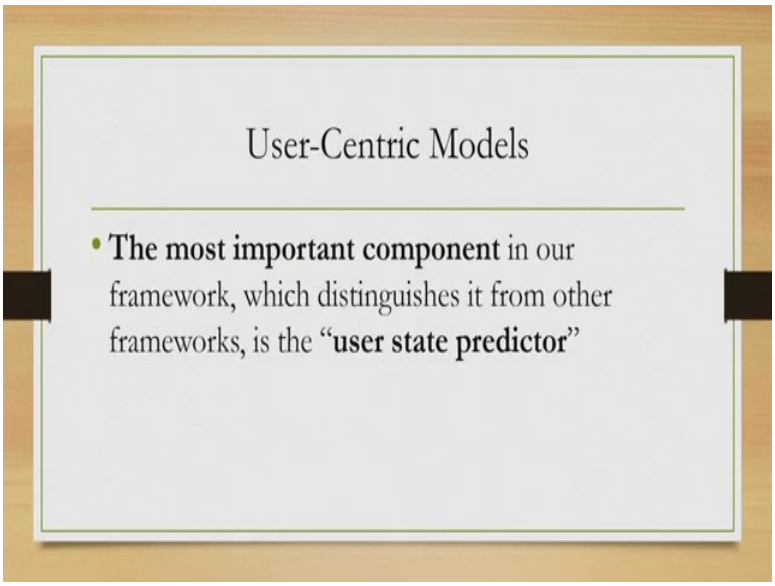
As you can see this is the framework that we have discussed in our earlier lecture. So, just to recollect there are this application layer or application software which has three sub layers. Sub layer 1 consists of three programs: collector program, input provider

program, and renderer program. Now the job of the collector program is to store the current state of the interface. The job of the input provider is to allow the users to provide input to the interface. And job of the renderer is to render the new state of the interface.

The state information collected by this program and the inputs that are logged by the input provider program are sent to the user state predictor, which is organized as sub layer 2 in our framework. And this user state predictor actually predicts the next desirable state of the user that makes the system perceived to be better. Now this next desirable state information is passed on to another predictor program which is interface that predictor, which then comes up with all sorts of specifications for the interface that leads the user to the next desirable state.

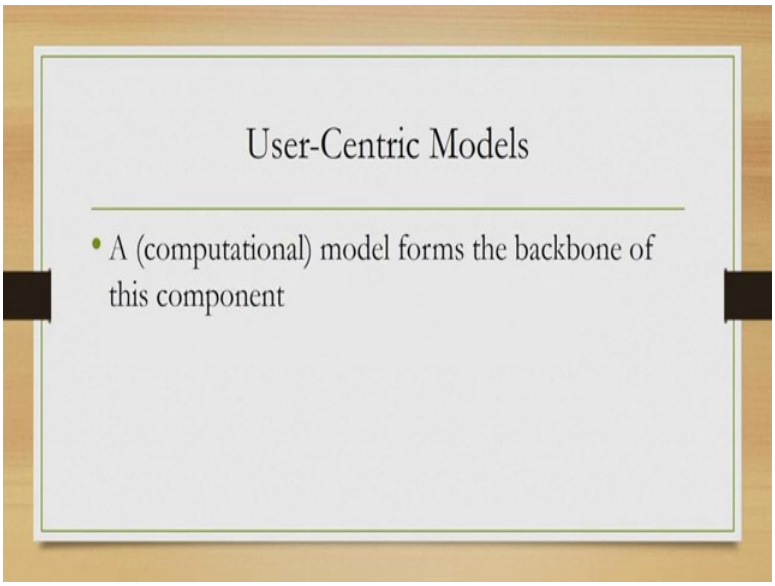
The user state predictor actually is the component that separates the software development lifecycle of an interactive systems from the other software developments. Now, this state predictor actually works based on the concept of a user model. So, in this lecture we are going to discuss in details the idea of user model with some user model specific user models as examples.

(Refer Slide Time: 03:48)



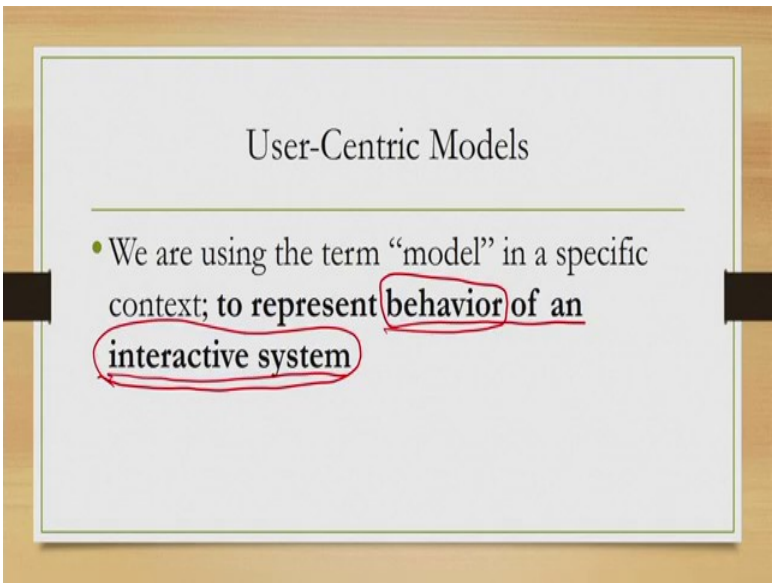
So, we can say that the most important component in this framework is the 'user state predictor'.

(Refer Slide Time: 03:56)



And the backbone of this predictor is the model. Now here, in the context of the framework we are talking of computational user model, but that is not the only type of user models that we have seen or that we used or that have been developed over the years. And we will have a introductory idea on different types of user models including some brief introduction to the computational models.

(Refer Slide Time: 04:24)

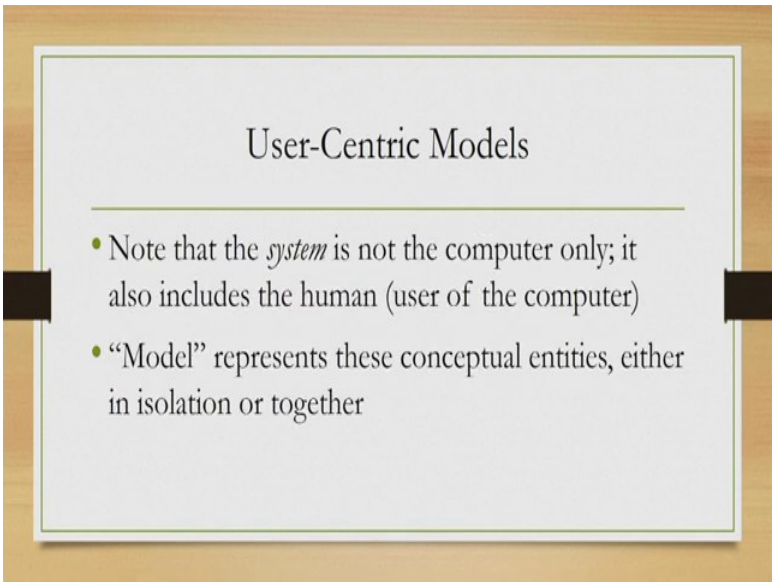


Now before we proceed further one clarification is required. We are referring or repeatedly using the term 'model'. Now this model what it refers to? So, when we are

using the term user centric models in the specific context of user centric systems, we are specifically referring to the behavior of an interactive system. Now note the emphasis given on the term behavior and interactive system.

So, when we are talking of an interactive system we do not talk of only the computer, instead when we talk of the system here our system is combination of human and the computer both.

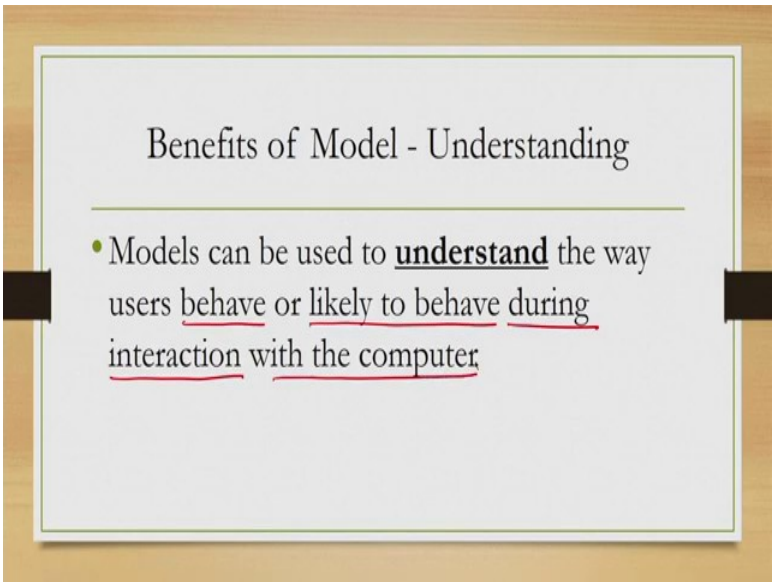
(Refer Slide Time: 05:10)



And the model represents these conceptual entities. Now sometimes the model may refer to the entities in isolation and sometimes it may refer to the entities together; both the user or the human and the computer. So, you have to keep in mind that whenever we use the term model we refer to either human or computer or both. So, it is an all-encompassing term.

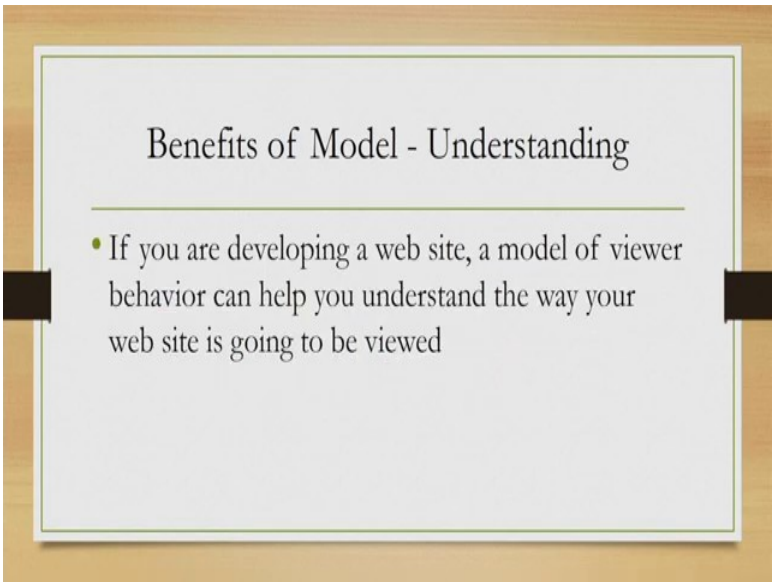
What we are going to do with the knowledge of a model. Before we explain the different types of models ok, we should know in details their use; what we can achieve with the idea of a model. Now models can help us in many ways.

(Refer Slide Time: 06:17)



The first thing is understanding. We can use the models to understand how the user behave or likely to behave during interaction with the computer. And that is helpful in many situations. Let us see with an example how it can help.

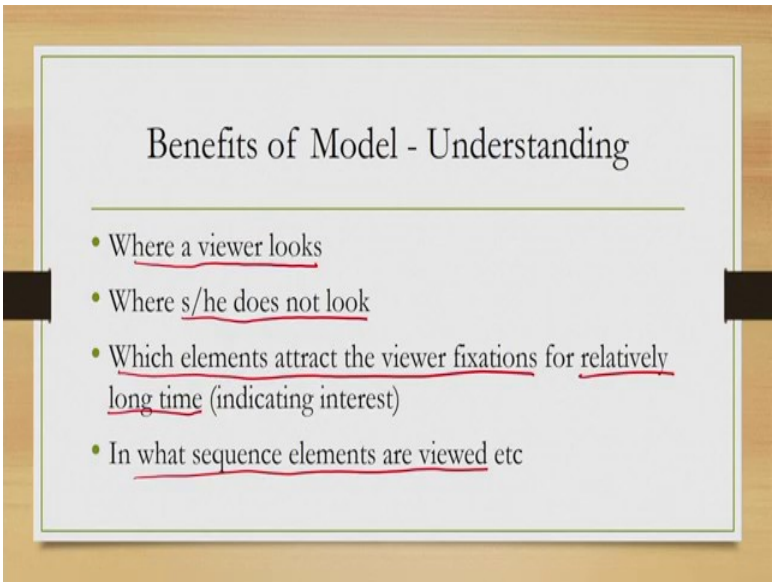
(Refer Slide Time: 06:40)



For example: suppose you are planning to develop a website and the website has many components. It may include something on the items that the company sell, it may include something on the stuff structure, it may include something on the job openings and many such things. It may even include some advertisement on the company web page.

Now, the point is when you are asked to design such a web page where there are many elements that is the user needs to be shown, how we were going to decide on which element to place where. So, the decision can be aided; the decision can be informed if you have a model of the way user gives a webpage.

(Refer Slide Time: 07:44)



If you have such a model that can help you understand where the user looks, where see he does not look, which elements attracts the fixations; the eye fixations for relatively long time. So, there may be elements where we may give a cursory look and move forward go to the next element, but there can be some elements were due to their design or due to their importance or due to the information content in those elements we may like to spend more time. So, a model of behavior can help you achieve or understand all these aspects of your design. Including in what sequence the elements are likely to be viewed.

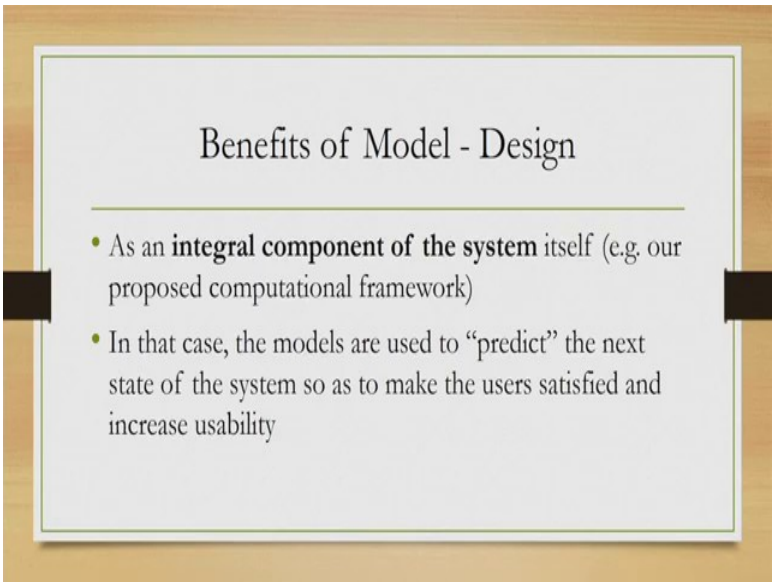
So, you will get to know with the help of a model before you start designing your website or you will get some idea on where a viewer looks, where is he does not look, which elements attract attention, which indicates relative interest, and in what sequence the elements are viewed. These are of course some of many such information that you can obtain from a model. That is the understanding.

Apart from that a model can also help you in design. So, the design guidelines; so in our earlier lectures if you may recall, we have discussed few guidelines including the Golden

Rules by Shneiderman. Now, these guidelines were derived based on some idea of the users based on the study of the users or in other words based on an implicit model of the user behavior.

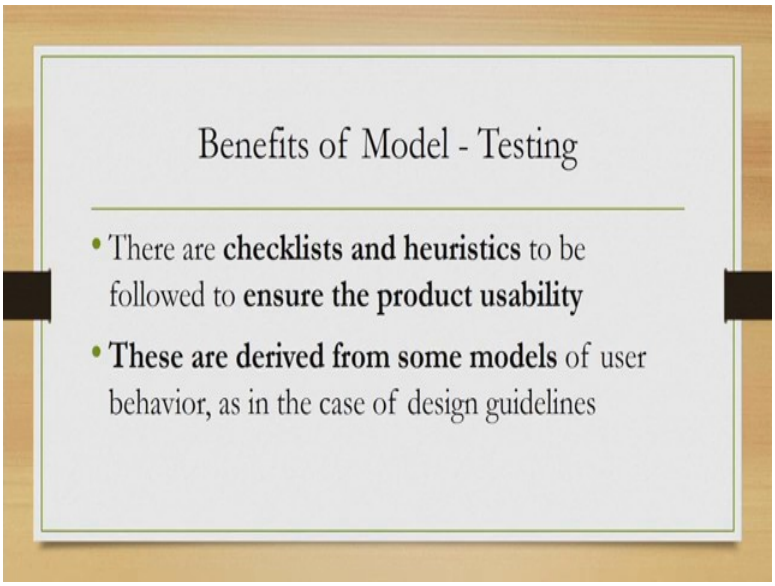
So, formulation of design guidelines is one advantage that you can get from a model.

(Refer Slide Time: 09:51)



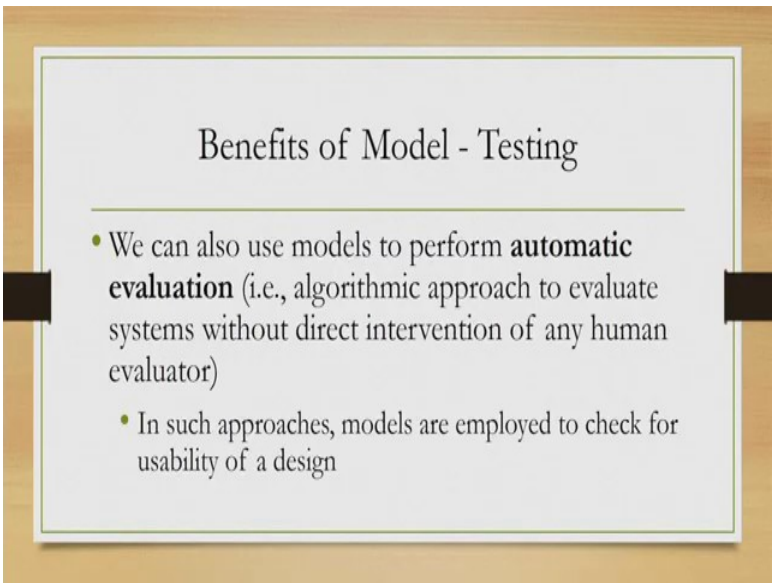
Otherwise also you can actually have the benefit of integrating a model into the system itself. Like we have discussed in our framework where we are planning to or we are looking forward to integrate a computational user model into the system itself; into the development process itself so that we can get a usable product with less effort and time. So, such models are computational or predictive as we will see in later discussions.

(Refer Slide Time: 10:32)



The third benefit that model can give us is in terms of testing your design. Now as we will see again later in a later lecture when we evaluate user centric system there are checklists, there are guidelines which we can utilize to evaluate our design. Now these guidelines or heuristics or checklists are all products of user studies and implicit user models.

(Refer Slide Time: 11:07)



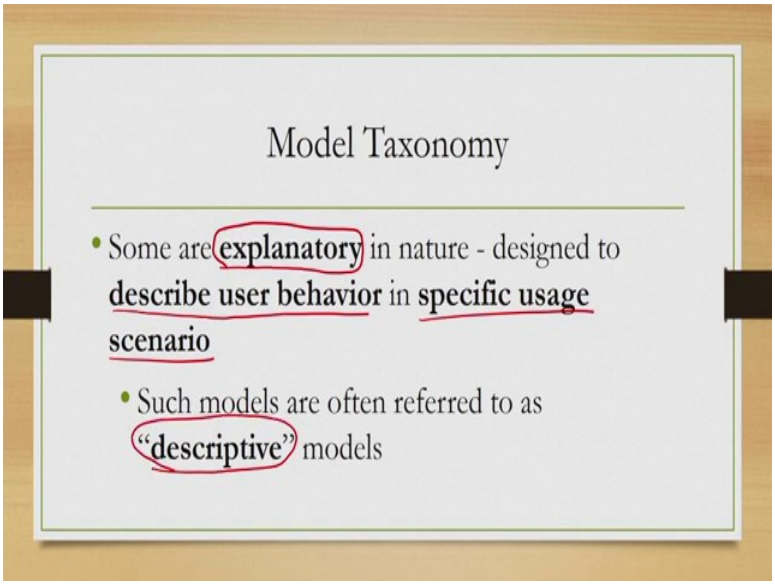
Another way of using a model in testing is basically to automate the evaluation process and to go for a model based evaluation. This topic will also be discussed in a later

lecture. But briefly just to give you some idea what this means is that; during evaluation we use a computational model to automatically evaluate the system or your design rather than going for a thorough user study detailed user study. So, that is another major advantage of having user models.

Now to summarize: so there are broadly three aspects of the entire design process that can be aided by the idea of a model. Namely, understanding of the actual design itself in terms of design guidelines or component of the design cycle as well as testing, where we can use checklists or heuristics to test which are derived from a model or we can automate the testing or evolution process of our design with the help of computational or predictive models.

So, these are almost all major components, these constitute almost all the major components in the software development lifecycle. And obviously, a single user model cannot be used to perform all these activities, what we need or what we have to perform these activities are a large number of such models. And all these models we can organize in the form of taxonomy.

(Refer Slide Time: 13:01)

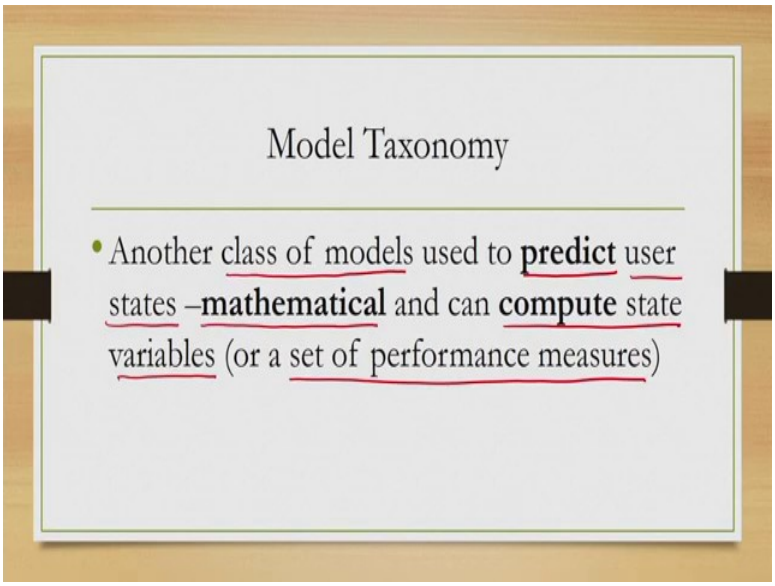


Now, some of these models are essentially explanatory in nature. So, when we use the term explanatory what we mean is that; these models can be used to describe user behavior in specific scenario. Apart from the description we cannot do anything, but the

description of course help us in understanding the design; understanding how to design it gives us some idea.

Since, these models are used to describe the behavior in specific scenario they are often called descriptive model. So, this is one category of model.

(Refer Slide Time: 13:49)

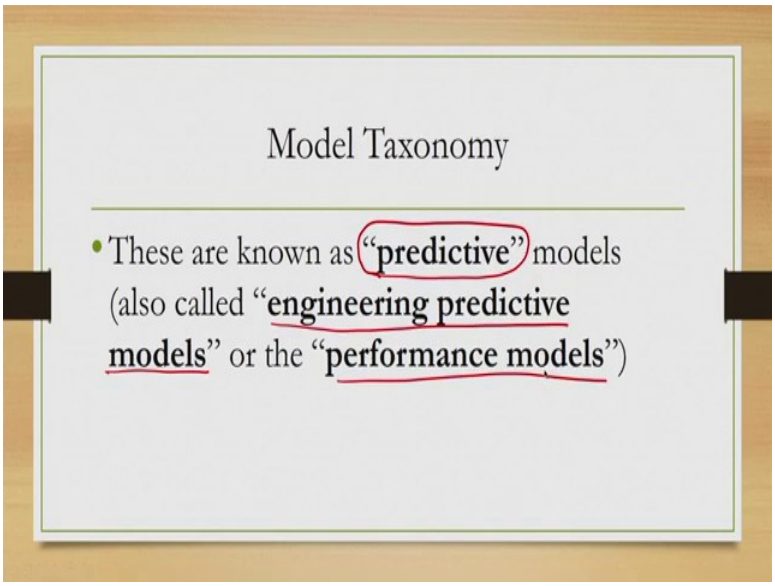


Next, we have another class of models which are used to predict the state of the user. Now if you may recollect from our earlier discussion when you say computation we mention that it is nothing but transition from one state to another, where the state is defined as a combination of system variables and or the computer variables and our mental variables.

Now, the set of mental variables constitute a user state and the class of models that we are talking of now are used to predict the states from the current state, what will be the next day what will be the values of the set of state variables. So, these models are essentially mathematical in nature and they can compute the state variables; as I have already explained.

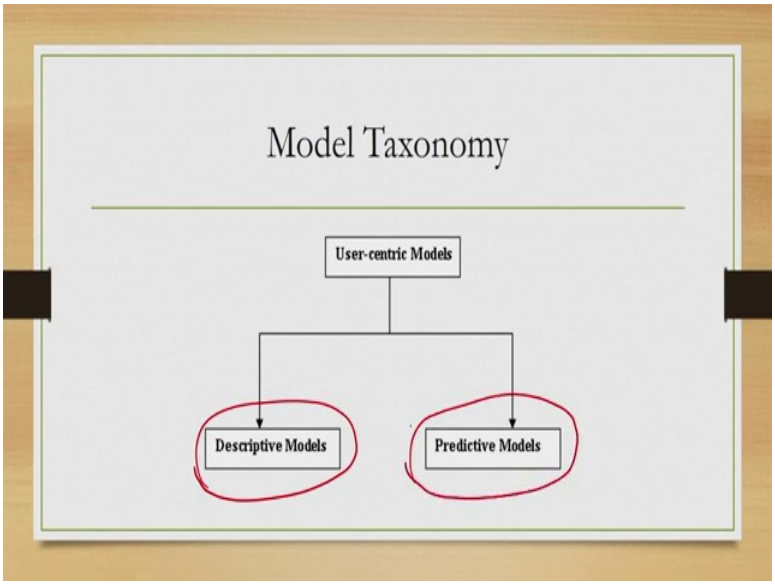
Now, these state variables as we have seen before can be a set of performance measures as well that are indirect indication of the user state.

(Refer Slide Time: 14:57)



Now, these models are known as predictive models. There are many other names by which they are known most popular ones are engineering predictive models or the performance models. So, predictive, engineering predictive models or the performance models are three of the most popular terms used to denote these class of models.

(Refer Slide Time: 15:27)

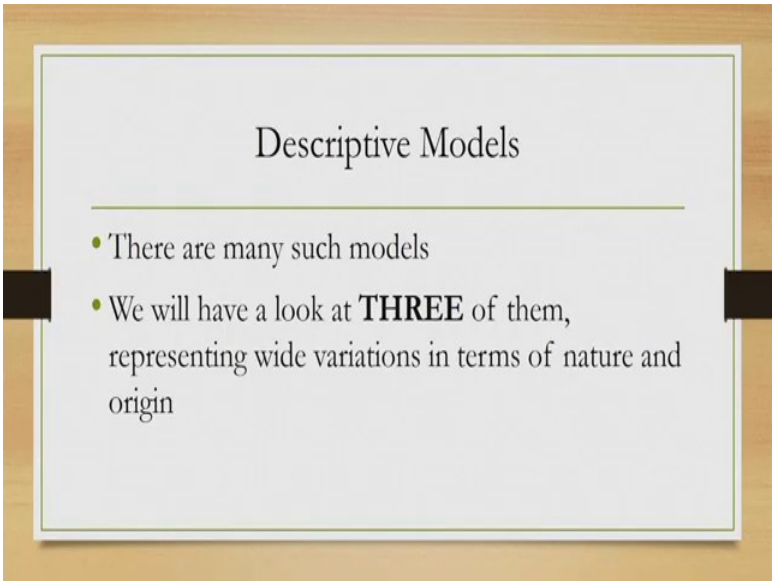


So accordingly, we can create a taxonomy of all sorts of models. So, user centric models you can broadly divided into descriptive models. So, this is one classic model which describes the behavior and predictive models, which computes the behavior.

Now let us briefly learn about the descriptive models although our focus in this course will be the predictive models. It is always beneficial to have a broad understanding of different types of models to appreciate the difference of the predictive models from the descriptive models and to appreciate or categorize a given model into one of these classes.

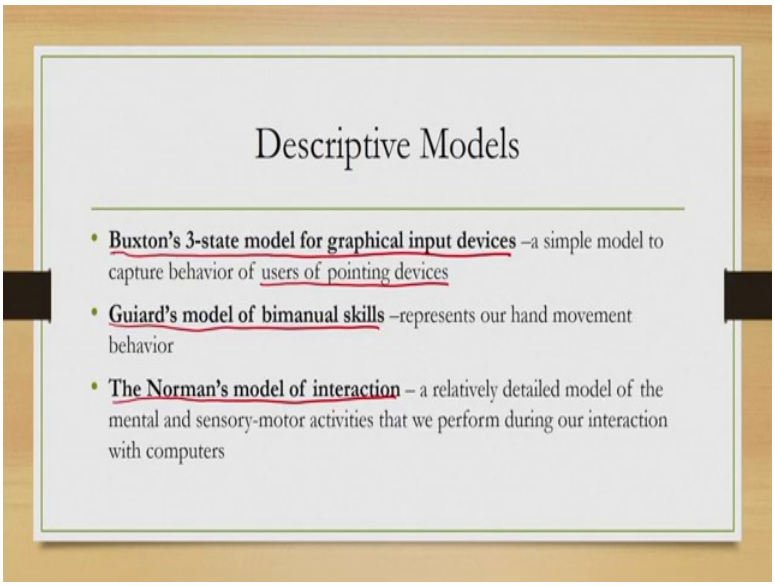
So, when we are talking of descriptive model just to recollect these are the models which are used to describe the behavior of a user in a specific usage context.

(Refer Slide Time: 16:36)



There are a large number of such models and their complexities vary from very simple model to very complex descriptive models. In this lecture we will have a look brief look on three of these models which are representative of the wide variation in complexity as well as their nature and origin.

(Refer Slide Time: 17:04)

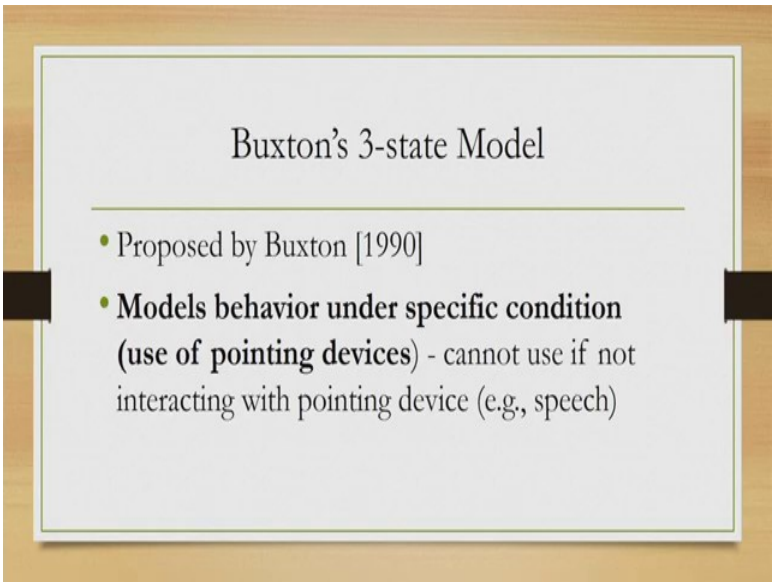


Now, these three models are the Buxton's 3-state model for graphical input device; which is a simple model to capture behavior of users of pointing devices. Now this is important and we will come back to this point later. Then we have Guiard's model of bimanual skills; which represents our hand movement behavior and the normal model of interaction; which is a relatively detailed model of the mental and sensory motor activities that we perform during our interaction with computers.

Now the three models: the Buxton's model is the simplest, Guiard's model is slightly more complicated than Buxton's model and the Norman's model is more complicated and detailed than the Guiard's model. So, in that way well get an idea of how these models differ in terms of complexity. Let us discuss these models one after another.

So, first is that Buxton's free state model. Although the name may sound long the model is actually very simple.

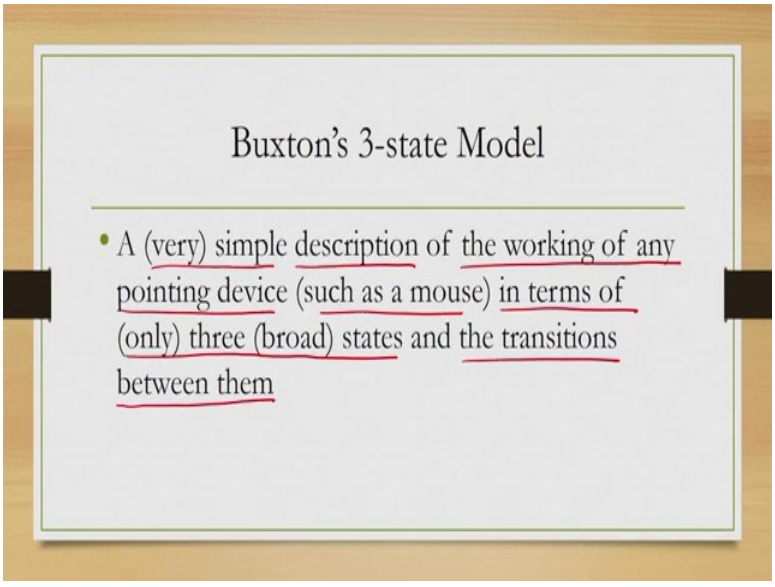
(Refer Slide Time: 18:08)



It was proposed by William Buxton in 1990. And it was proposed to model specific behavior under specific conditions that is the use of pointing devices. This model is meant only to describe the behavior of pointing devices, it cannot be used to describe the behavior of users when they are using any device that is not a pointing device. And that has to be kept in mind clearly.

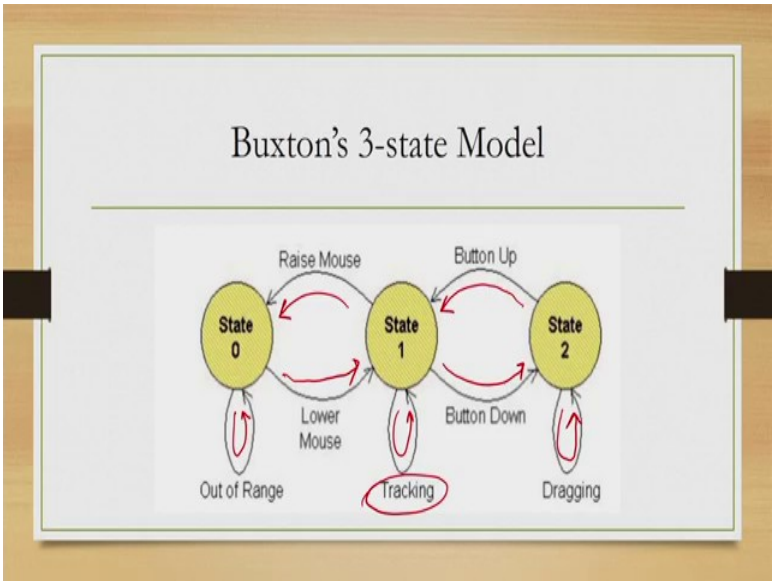
For example: if somebody is interacting with the system using speech, we cannot use this model to describe that behavior.

(Refer Slide Time: 18:55)



And as I have repeatedly said the model is: a very simple description of the working of any pointing device such as a mouse in terms of only three broad states and the transitions between them. Now these states again, although I am using the term state here there will be some difference between the states we have discussed earlier in the context of explaining the idea of computation and the state we are discussing here. Yet, the states are very simple and explanatory rather than very detailed with a set of state variables that model the system.

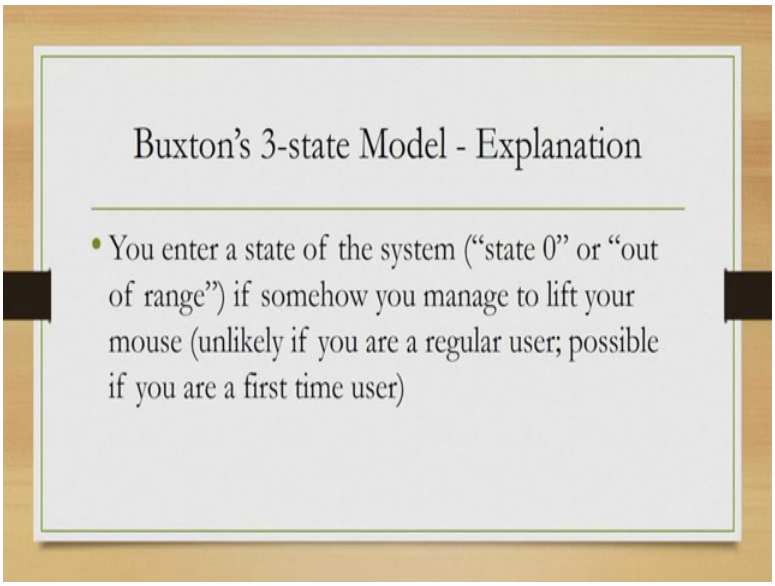
(Refer Slide Time: 19:41)



So, this is the model. So, here we have 3-states named as: state 0, state 1, and state 2. Note here that these states we are not defining for these states any state variables we are simply naming them. So, it is kind of explanatory states rather than or descriptive strategies rather than state in the context of computations. And when we are talking of these 3-states along with the states we are talking of transitions between states. The transitions are shown in these arrows. So, there are total seven transitions including self loops. So, there are four transitions between the 3-states and three self loops.

So, let us explain what this model explains.

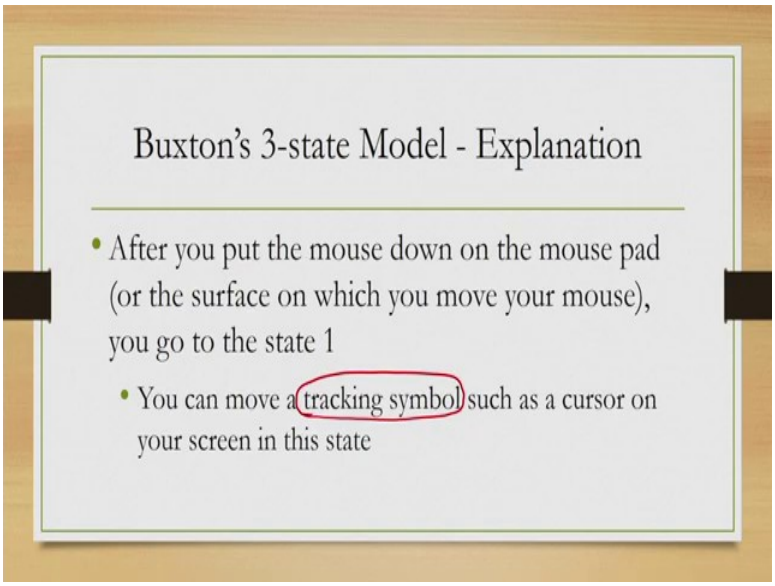
(Refer Slide Time: 20:33)



Now, suppose you are interacting with the mouse, you are using the mouse to interact with your computer. Then if you are a first time user there is a possibility that you may lift the mouse from the surface although it may sound very improbable, but sometimes it happens. So, by mistake if you do that or without knowing if you do that then you go to state 0 in the model where it is defined as out of range state. Now you remain in this state as long as you keep the mouse above the surface or not in touch with the surface.

So, once you lower the mouse, once you put it down on the surface then there is a state transition you come to state 1.

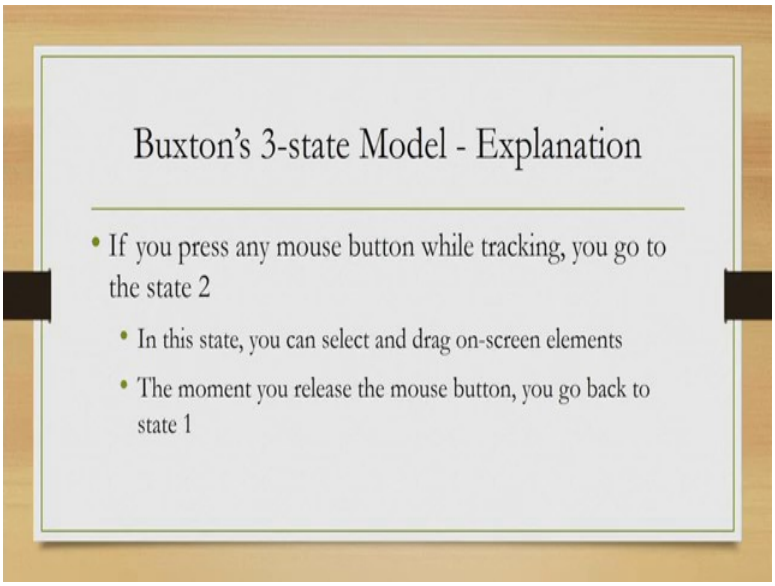
(Refer Slide Time: 21:34)



Now in state 1, what you can do is basically you can track a cursor or in more general terms there is a tracking symbol which you can track it. In mouse typically that is the cursor that you track. So, as long as you are using the mouse for tracking then you are in state 1. And in state 1, if you again by mistake lift your mouse, then you go back to state 0 and the moment you put it down you come back to state 1. And in state 1 if you keep on tracking then there will be the self loop of tracking.

Now in state 1, if you press a button which is denoted as button down then you go to state 2.

(Refer Slide Time: 22:24)



So, what happens in state 2? In this state by pressing the button you can actually select an object and you can drag it. So, this selection and dragging as long as you do this. You are in state 2 and the moment you release the button you go back to state 1.

So, to recollect what this model tells us is that; you want to use your mouse or a pointing device to interact with the computer and your hand is on the point in device. Now, by mistake you lifted it out of the surface on which it should be dragged. Then you are in step 0 or at the beginning also when you are not doing anything you are in state 0. And in this state as long as there is no touch with the surface on which it should work it will remain in state 0; the moment you touch the surface with the mouse and start using the mouse or start dragging the mouse on the surface you come back to state 1.

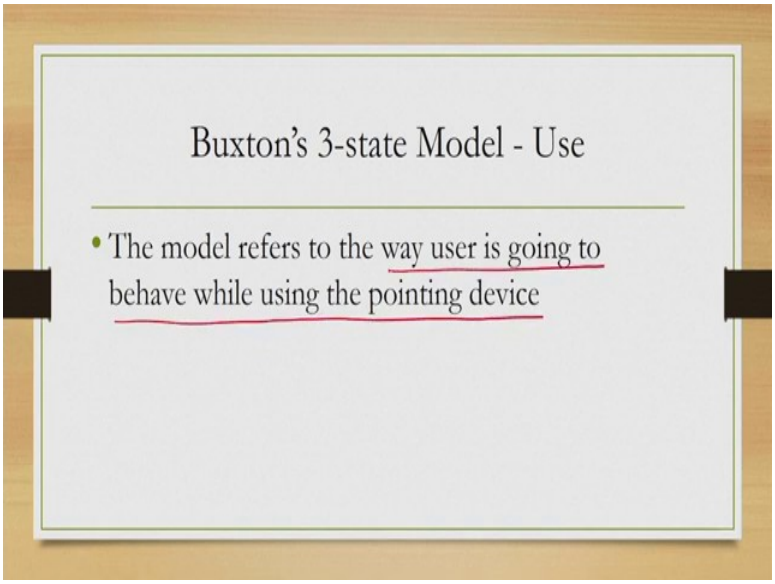
In state 1 you can actually move the pointer on the screen which is called tracking. This is the one of the two major activities with any pointing device and as long as you keep on tracking. So, through the self loop, you remain in state 1. In state 1, if you again by mistake raise the mouse then you make it state transition to state 0. And as long as the mouse is lifted through the self loop you remain in state 0; the moment the mouse or the tracking device touches down the surface you come back to state 1.

In state 1 if you press a mouse button then you go to state 2. In this state you can actually select and drag on screen elements and as long as you are doing that you remain in state

2 as explained by the self loop. And the moment you release the button go back to state 1. So, that is another state transition from 2 to 1.

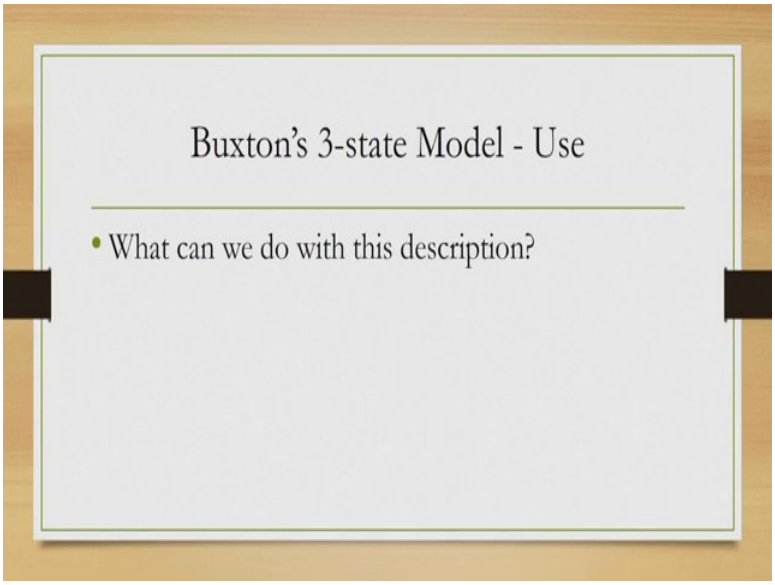
You may be wondering what is so special in this explanation.

(Refer Slide Time: 24:52)



It is a way to refer to the behavior of a user who uses a pointing device. So, the model refers to the way user is going to behave while using the pointing device.

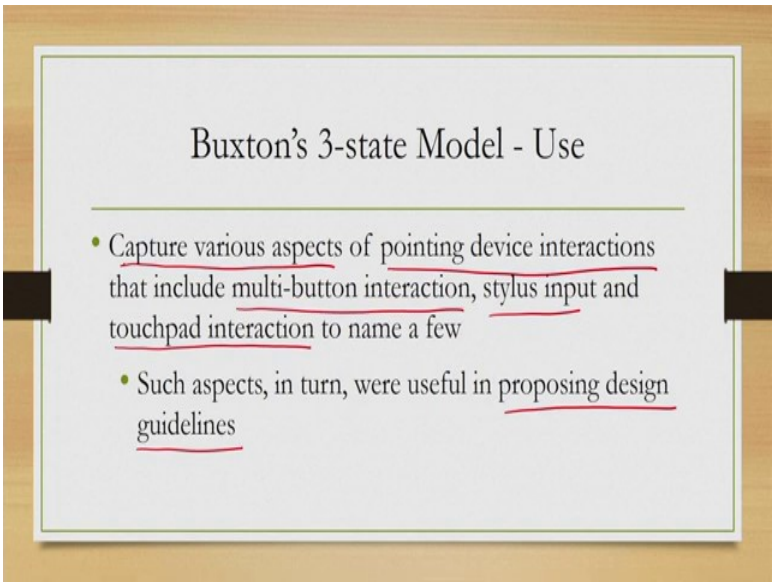
(Refer Slide Time: 25:09)



Now, this model is of course very simple and you may be wondering that so what; that is intuitively what we understand. And it is most likely to be obvious to most of us that when a pointing device is given we behave in this way, but that is not the case. Actually there are 3-states, but when we think of intuitively we think of two states. In this context state 1 and 2, so we never bother about state 0 because most of us fails to realize that it is also an important thing to consider while designing appointing best interaction. And if we do not consider it then the interaction experience may not be as good as when we considered that state.

So, essentially this gives us some idea of how to design an interaction when we are using a pointing device by considering all the 3-states instead of two. And this is another way of structuring our third process about the interaction with the pointing device which helps us in many ways. So, with this simple understanding of the pointing based interaction; pointing device based interaction with the computer we can do a lot of things.

(Refer Slide Time: 26:35)



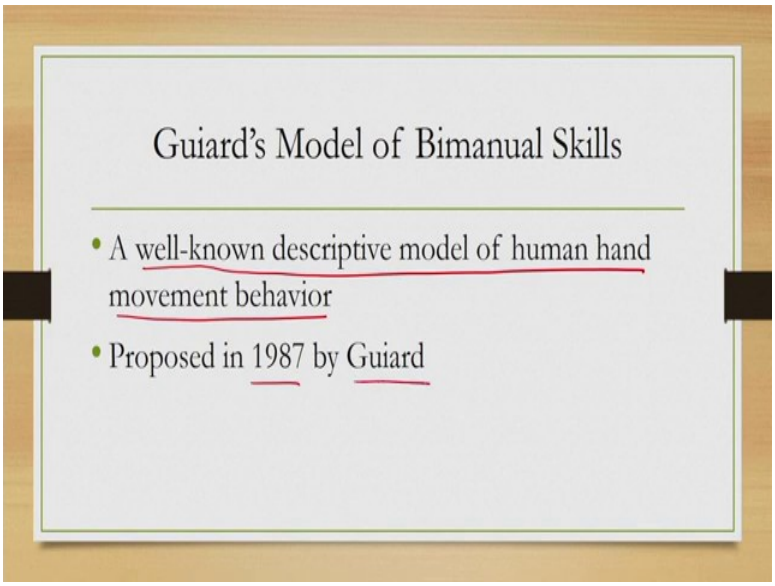
And although this is not the right place to go into the details of all those things that have been done all those knowledge that have been gained from this simple model. Briefly what we can say is that the model has been successfully used to capture various aspects of pointing device interactions that include multi-button interaction, stylus input, and touchpad interaction.

But these are of course only a few, this is not an exhaustive list and there are many other such inter actions that benefited based on the idea gained from this simple model. And these ideas that we can obtain for interactions involving these type of devices can help us in proposing design guidelines. That is another important consideration. That once we capture the knowledge; once we capture various aspects what to do with that.

So, in this case with this descriptive model of user interaction through a pointing device we can actually convert this knowledge to a set of guidelines design guidelines. How to design? An interface where the user will interact through a pointing device in a better way so that all these states are taken care of.

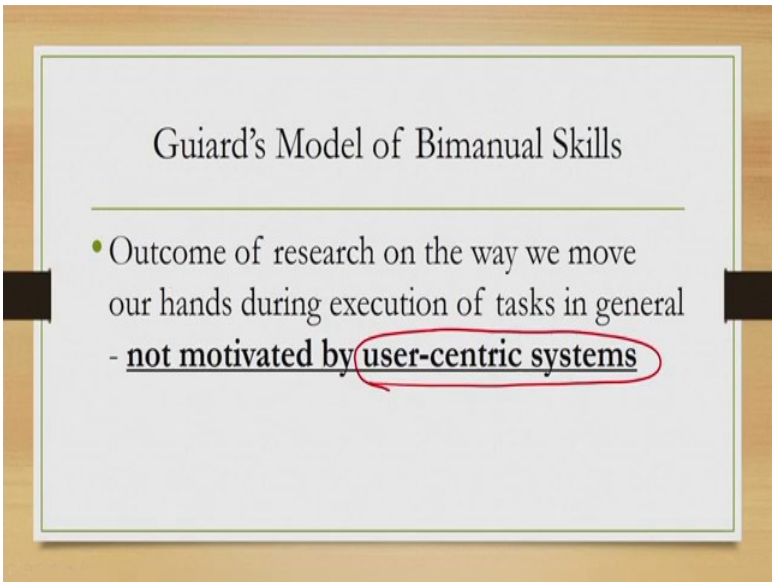
Next, let us consider a slightly more complicated model that is called Guiard's model of bimanual skills.

(Refer Slide Time: 28:18)



Now, this is a very well-known descriptive model of human hand movement behavior. So, this is a model that helps us understand the human hand movement behavior. And it was proposed way back in 1987 by Guiard.

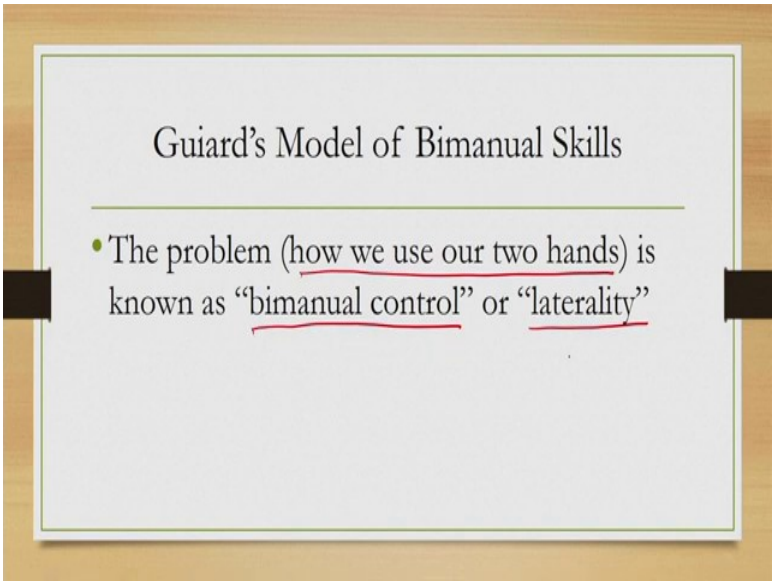
(Refer Slide Time: 28:45)



An important thing to note here is that the model was not proposed for use in user centric systems. So, that was not the original intention of the researcher. It is generally the model was the outcome of research on; the way we move our hands during execution subtask in general. So, there is no specific constellation of interactive systems and the way we move our hand in the context of interactions with a computer.

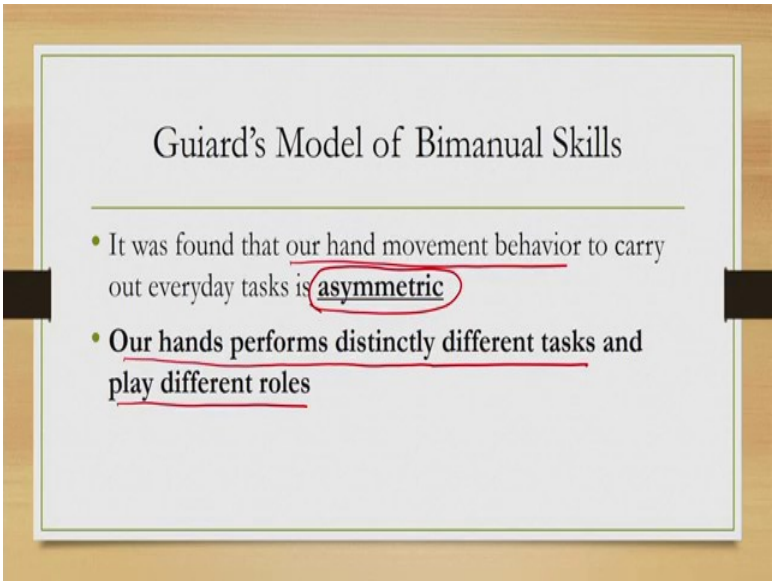
So, this is an example of a model that was developed for a broad problem which found its use in a specific context that is interaction with a computer.

(Refer Slide Time: 29:38)



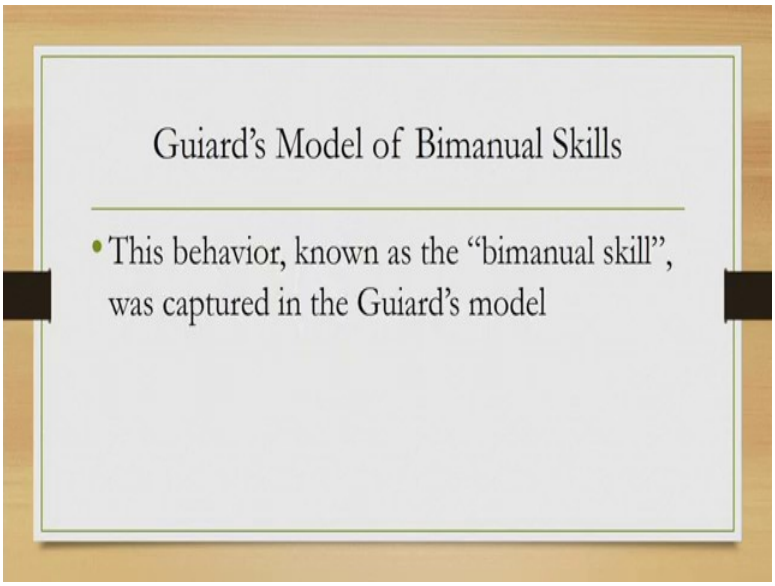
So, the problem is how we use our two hands. We need to know that is our problem. Now this problem is broadly known as the bimanual control or laterality. So, how we use our two hands is a problem that is more popularly known as bimanual control or laterality. So, what has been found in the research is that we do not use our hands both the hands in the same way. So, there is some asymmetric usage.

(Refer Slide Time: 30:03)



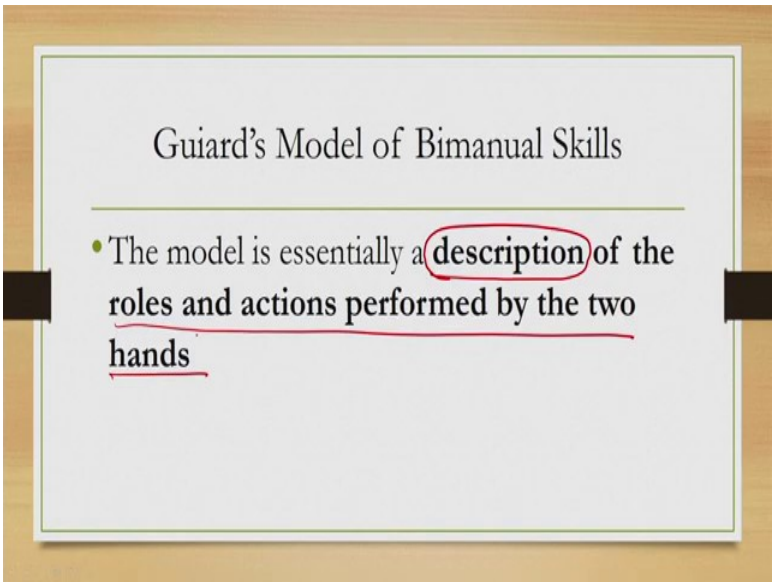
So, our hand movement behavior is asymmetric. So, when we execute a task our hands perform distinctly different tasks and play different roles, which is not generally what we think of.

(Refer Slide Time: 30:36)



Now, this behavior that our hands perform different roles and perform different tasks during execution of some activity is known as the bimanual skill which is captured in this particular model by Guiard.

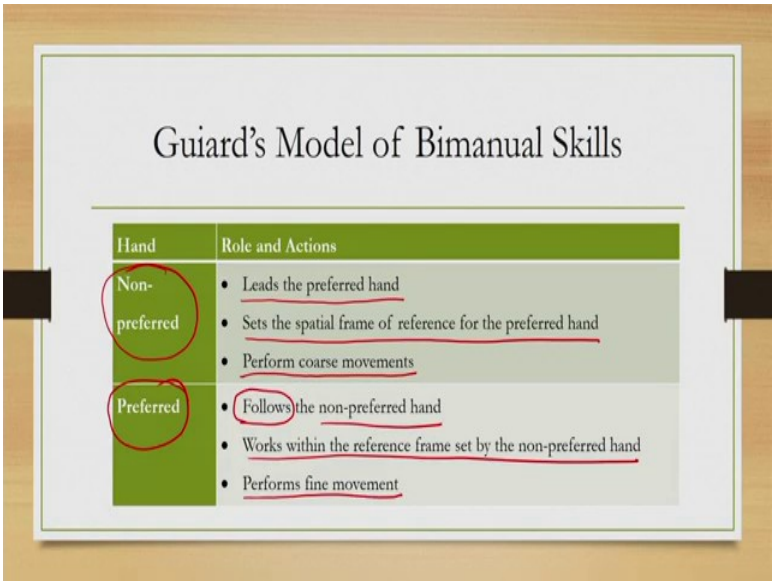
(Refer Slide Time: 30:58)



So, when we were talking off the model just to repeat again here it is essentially a description of the roles and actions performed by the two hands. So, since it is only a description of these things, so we are calling it a descriptive model.

So, what it does it tells us; what roles and what actions are performed by different hands during the execution of a task.

(Refer Slide Time: 31:27)

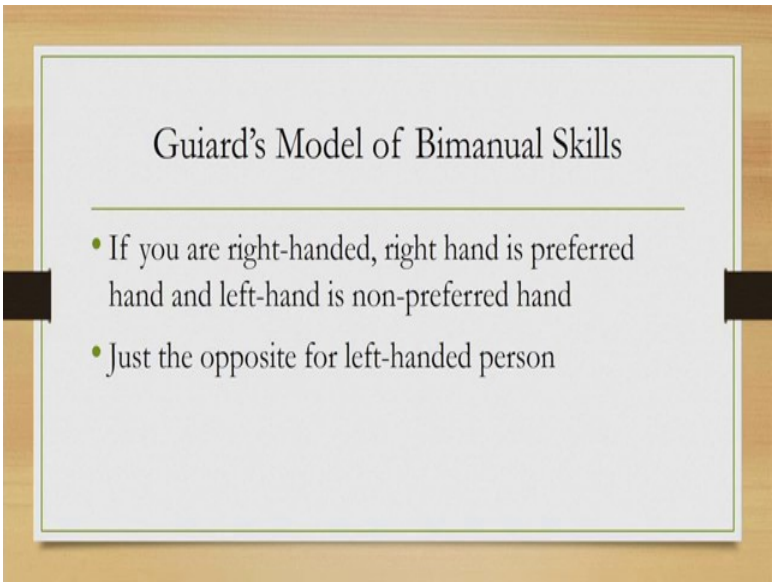


Now, something interesting is there. The model tells us that we can divide our hands into two categories: one hand acts as preferred hand, other hand acts as non-preferred hand. And the roles and actions are different for each of these.

So, when you are talking of non-preferred hand it performs three roles and certain actions. So, it is actually the task of the non-preferred hand to lead the preferred hand; it sets the special frame of reference for the preferred hand. In other words it sets a region in which the preferred hand is likely to move around and it performs some infrequent and small movements.

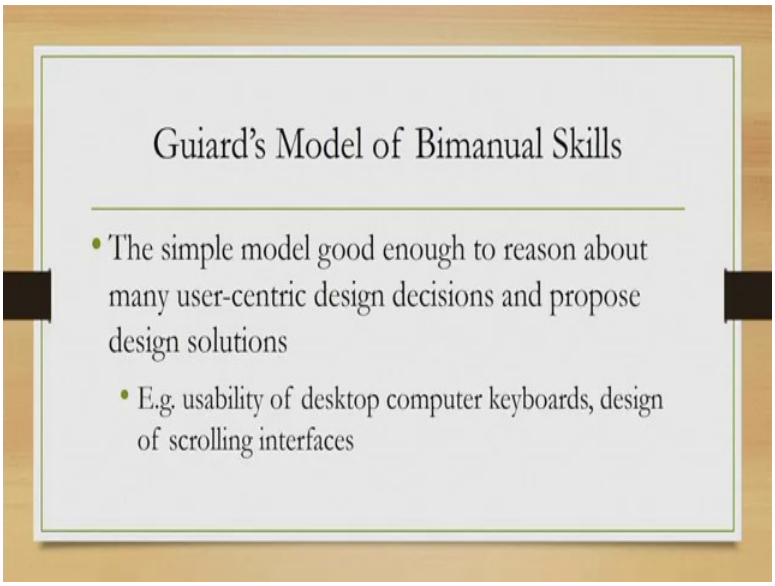
When it comes to preferred hand, this preferred hand follows the non-preferred hand and works within the reference frame said by the non-preferred hand and performs finer movements. So, more continuous and large movements

(Refer Slide Time: 32:48)



Now, what is the idea of this preferred and non-preferred hand? If you are a right-handed person then your preferred hand is the right-hand and left-hand is your non-preferred hand. And if you are a left-hander then it is just the opposite.

(Refer Slide Time: 33:03)



So to recollect, suppose you are a right-handed person so your preferred hand is the right-hand and your non-preferred hand is the left-hand. So, intuitively when we perform some task with our right-hand we assume that there is no role played by the left-hand.

But what this model tells us is that that is not the case, both preferred and non-preferred hands have roles to play.

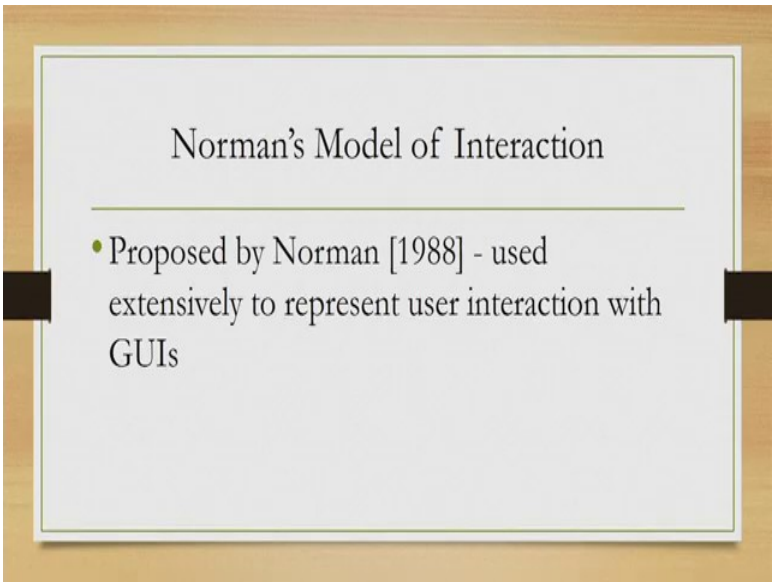
The non-preferred hand actually leads the preferred hand and sets a region of activity movement and perform some small movements; discrete small movements whereas, the preferred hand performs the actual activities or longer and continuous moments, big moments. But it remains within the region set by the non-preferred hand and it follows the non-preferred hand.

So, the simple knowledge of how our hands move actually helped in coming up with design decisions and design solutions that are useful in making a product usable. For example: the knowledge is used for better design of computer keyboards scrolling interfaces and so on.

Our third model is a slightly more complicated than the previous bimanual skill model. We are using the term slightly more complicated with respect to the idea that it is a bit more detailed; compared to the detailed information that is provided in the Guiard's model.

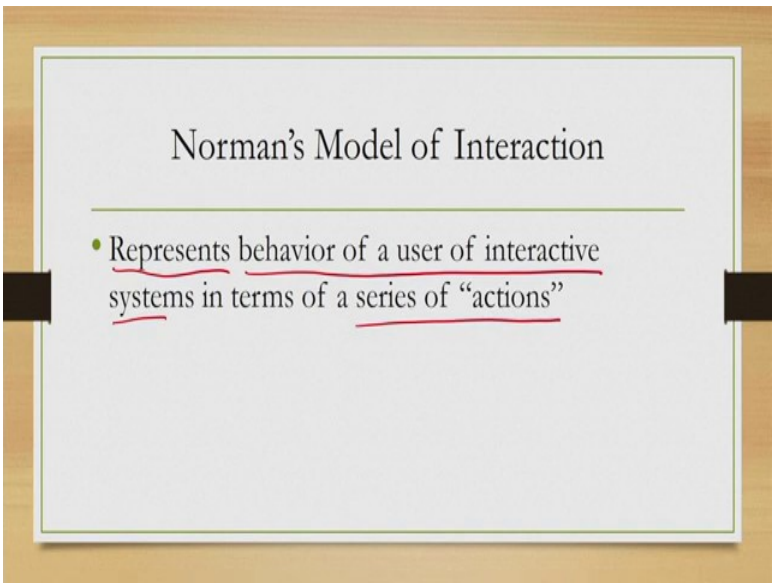
So, if we recollect the Buxton's 3-state model is the simplest because its details are not much; compared to that the Guiard's model is slightly more detailed. And now the third model that we are going to discuss the Norman's model of interaction will be even more detailed compared to the Guiard's model.

(Refer Slide Time: 35:36)



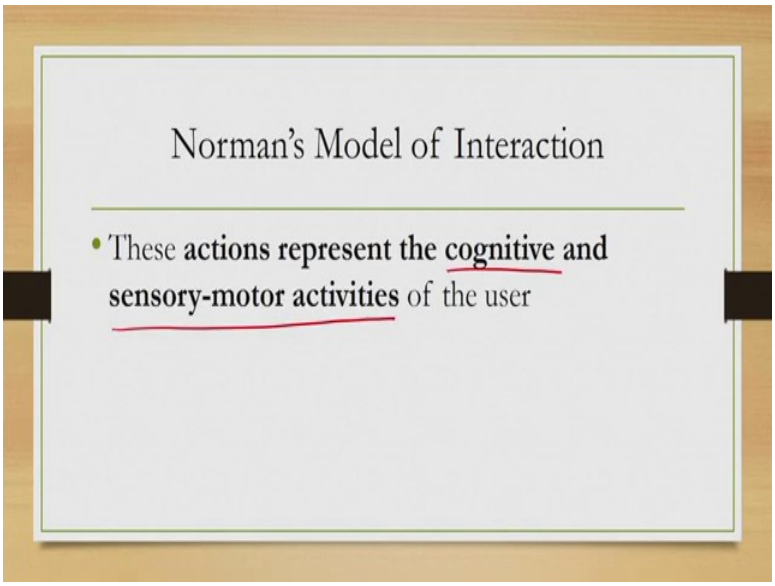
Now, this model was proposed by Donald Norman long back in 1988, and the model was used extensively to design GUIs and represent various user interactions with GUIs.

(Refer Slide Time: 36:00)



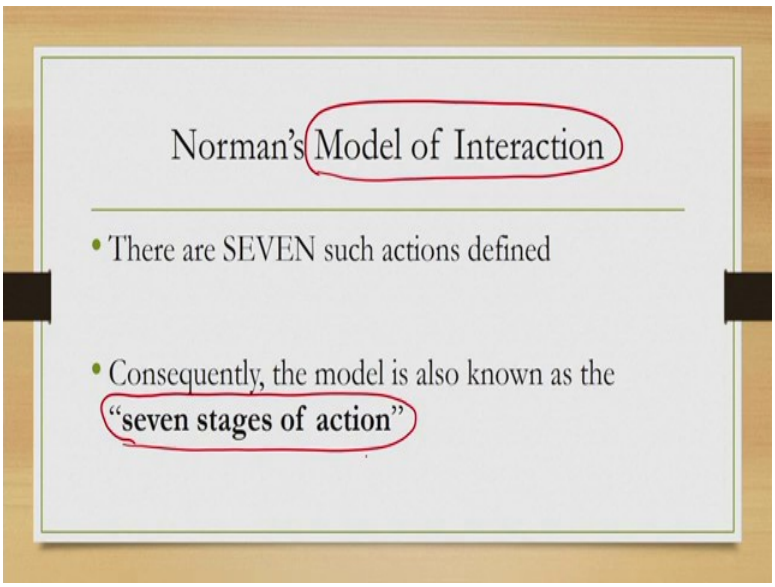
What this model does? It represents behavior of a user of interactive system in terms of a series of actions. So, note the emphasis given on these words: it represents behavior of a user of interactive system in terms of a series of actions.

(Refer Slide Time: 36:28)



What these actions mean? Now, if you can recollect when you talked of an interactive system we talked off a combined entity both user and computer together. Now these actions essentially referred to the cognitive and sensory motor activities of the user. So, this model actually helps us in a way to represent these activities that are part of the users rather than that are done by the computer.

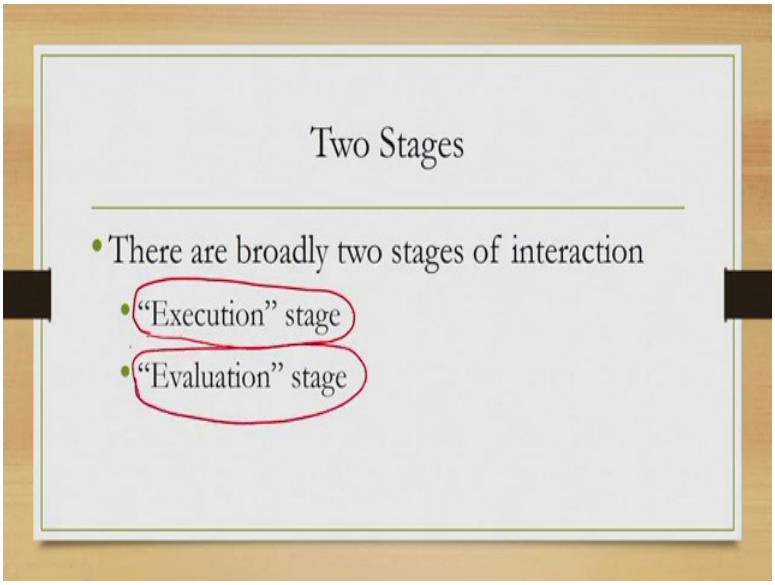
(Refer Slide Time: 37:15)



Now according to this model the model of Norman there are seven such actions. Accordingly, sometimes this model is also known as the seven stages of action. So, we

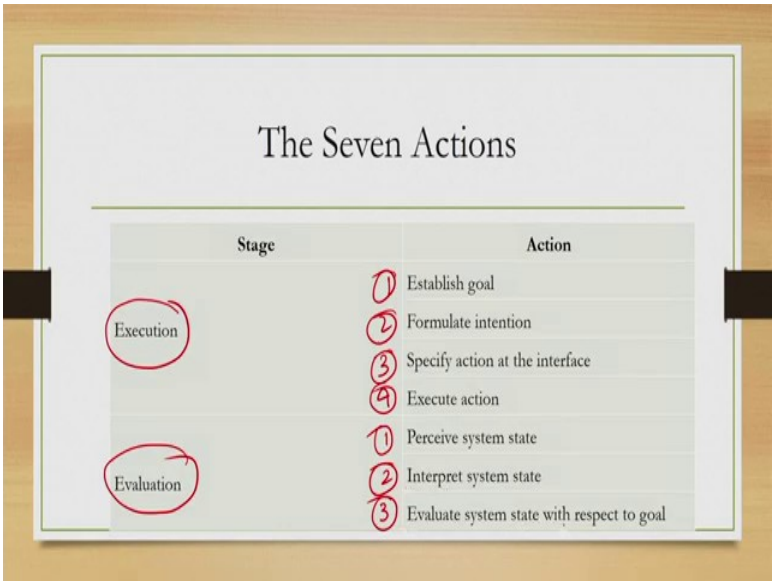
can use both the terms either Norman's model of interaction or seven stages of action, both are used.

(Refer Slide Time: 37:45)



Now, the seven actions that are there that are defined as part of the model can broadly be divided into two groups: the execution stage actions related to the execution stage and actions related to the evaluation stage.

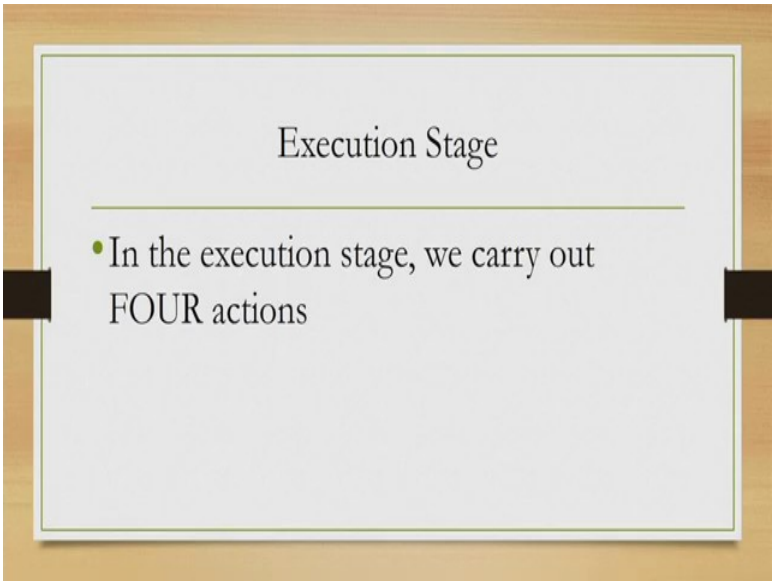
(Refer Slide Time: 38:07)



So, these are the seven actions divided into two groups: execution stage, there are four actions action 1, 2, 3, and 4; in the evaluation stage again there are three sections: 1, 2, and 3.

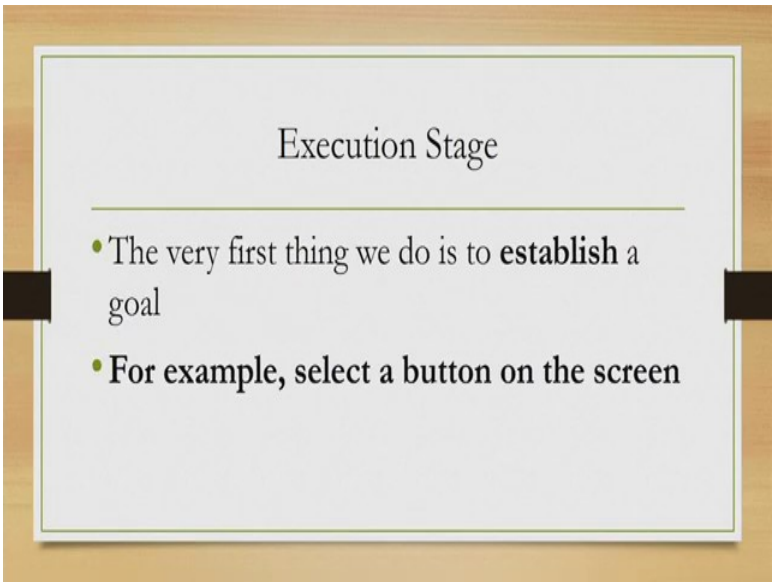
So, action one is establish goal, two is formulate intention, three is specify action at the interface, and four is execute action: under execution stage. Under evaluation stage: first action is perceive system state, that second action is interpret system state, and third action is evaluate system state with respect to goal. Together there are seven such action divided into two groups.

(Refer Slide Time: 39:02)



Now, let us try to understand one by one what these actions mean. So, as I have shown under the execution state there are four actions.

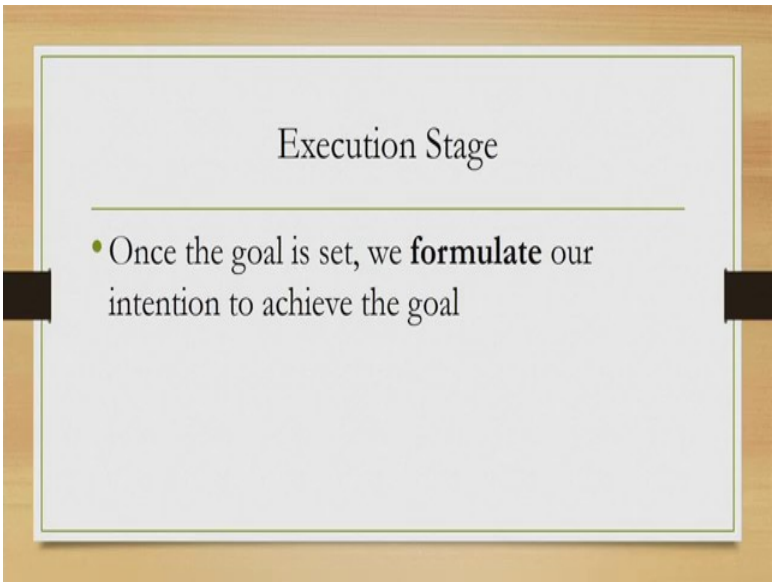
(Refer Slide Time: 39:14)



The very first action is establish a goal. Now for example, suppose you are asked to click on a button. So, clicking on the button is your goal; so you have to establish this goal first. To make it a bit more simpler rather than clicking on the button let us think of selection of a button which eventually will lead to of course clicking, so selection is a (Refer Time: 39:44) for clicking. But to make it simpler for us to understand so let us tick to the goal of selection of a button.

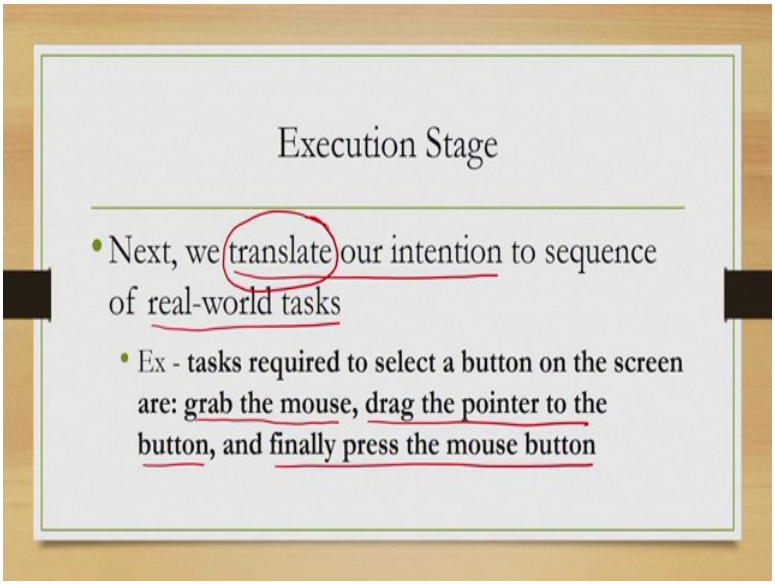
So, first action in the execution stage is setting a goal in this case it will be select a button on the screen.

(Refer Slide Time: 40:03)



Now, once the goal is set formulate our intention to achieve the goal. So, this is the next stage. So, mentally we do certain things that is called formulation of our intention; that in order to select the button we have to do these that, after these that something. Some intention that I have to first do these, after that these, that is mental thing. One way to understand is that to select a button you may have to mentally decide to identify the button then do something to select it; that is our intention.

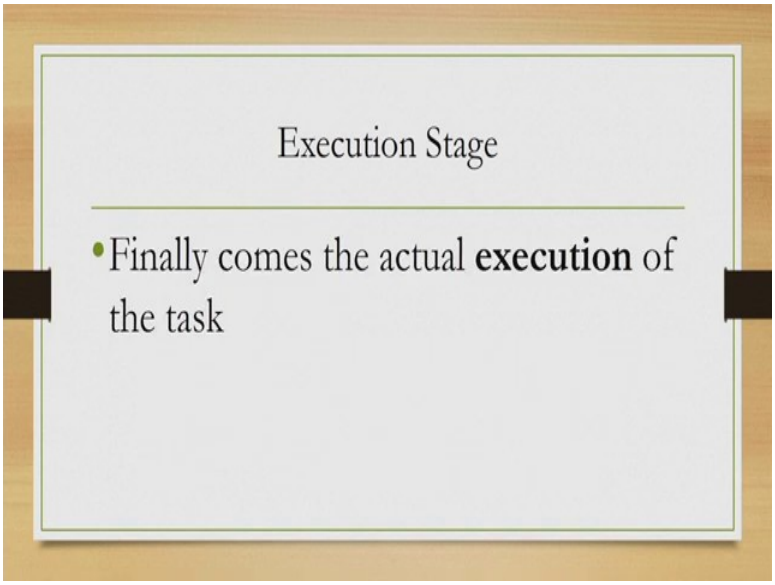
(Refer Slide Time: 40:46)



Once that intention is formulated we translate our intentions into actions or sequence of actions that are supported by the system. So for example, in order to select the button what are the actions we need to do on the screen. That is grab the mouse, drag the pointer and finally place the mouse button. So, these three things: grab the mouse, drag the pointer to the button and finally press the mouse button these are real world tasks that we need to do to translate our intention to achieve the goal.

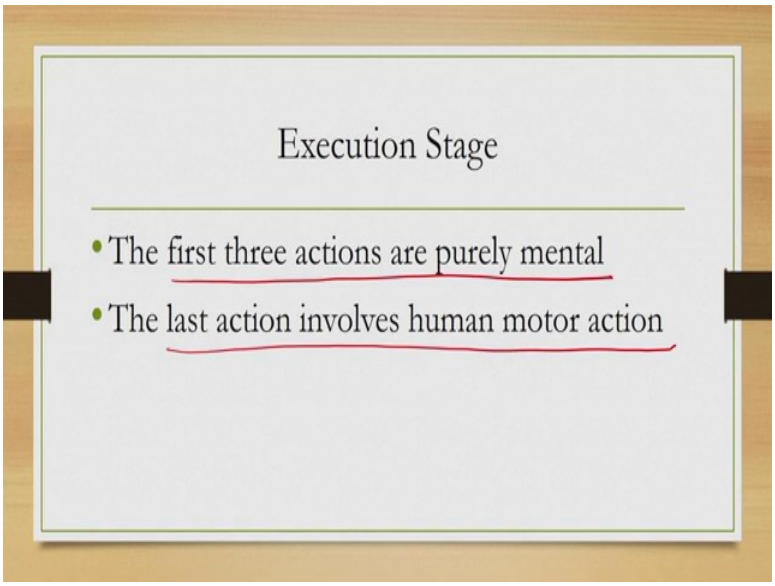
So, the third action in the execution stage is translates enough intense into a sequence of real-world tasks.

(Refer Slide Time: 41:39)



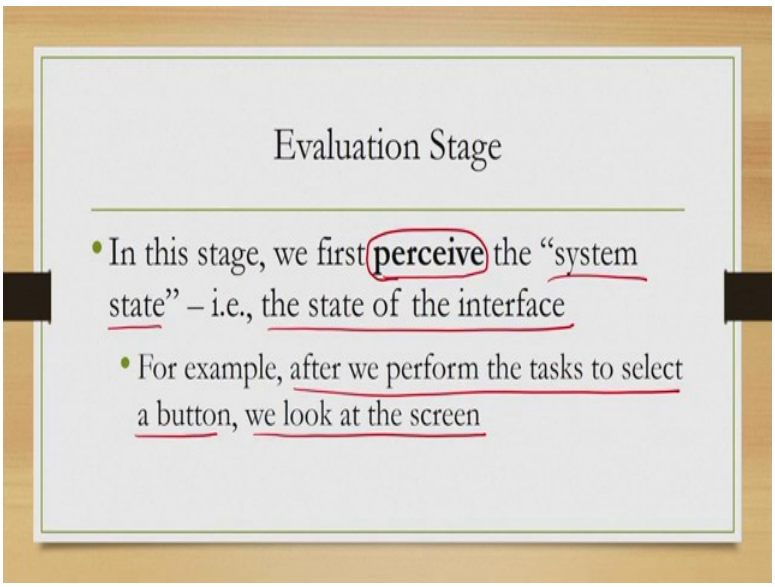
And finally, we execute the task. The set of tasks that we just mentioned before that to select the button the setup tasks involve grabbing the mouse, dragging the pointer and clicking. So, that setup tasks we execute and that is the final action in this execution stage.

(Refer Slide Time: 42:09)



Now among these four actions the first three are purely mental. As you may have probably noticed, so when we say that set a goal formulate intention and translate the intention to real world actions these are done in your mind and when you are asked to execute those actions on the screen that involves human motor activities. So, that is a motor action. So, that is the execution state.

(Refer Slide Time: 42:41)

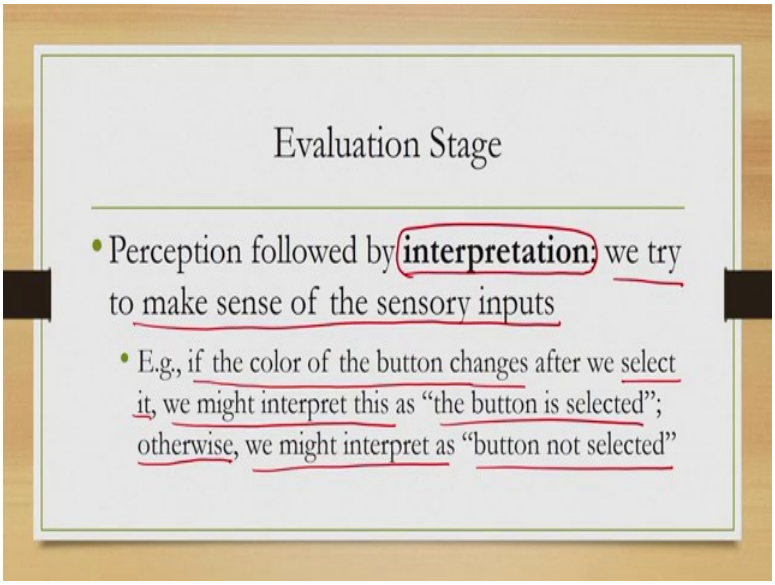


Next comes the evaluation stage. In this stage we have three actions. Now let us take up these actions one-by-one. So, what is the first action that we first perceive the system

state. So in the evaluation stage, the first action is that perception perceive the system state or the state of the interface.

To continue with our earlier example; so after we perform the tasks to select a button we look at the screen to perceive any changes on the screen. So, the activity of looking at the screen is perception.

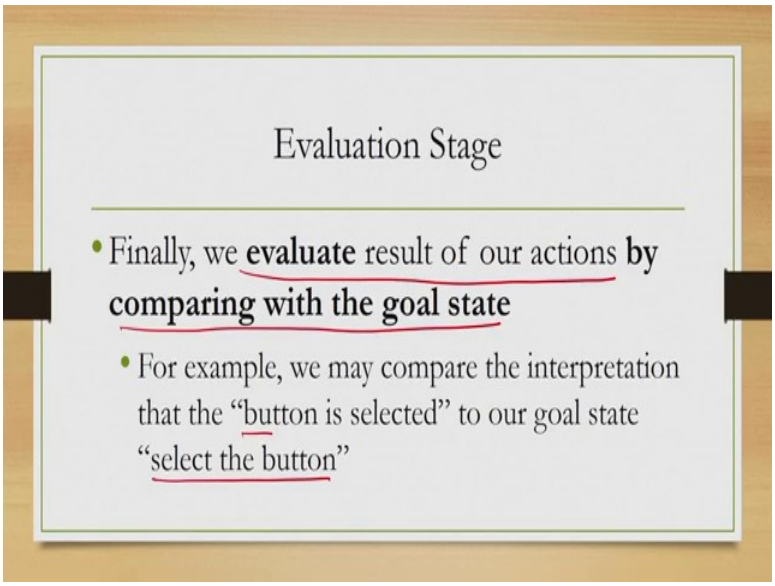
(Refer Slide Time: 43:36)



The perception is followed by interpretation. So, what we see on the screen or on the interface and we try to make sense of that sensory input. So, perception is essentially a sensory input. Now perception maybe visual maybe auditory maybe both. Here of course, in this example it is only visible, but sometimes it meanwhile auditory input also. So, these are input to our perceptual systems and that input is processed to interpret what those perceptual input means.

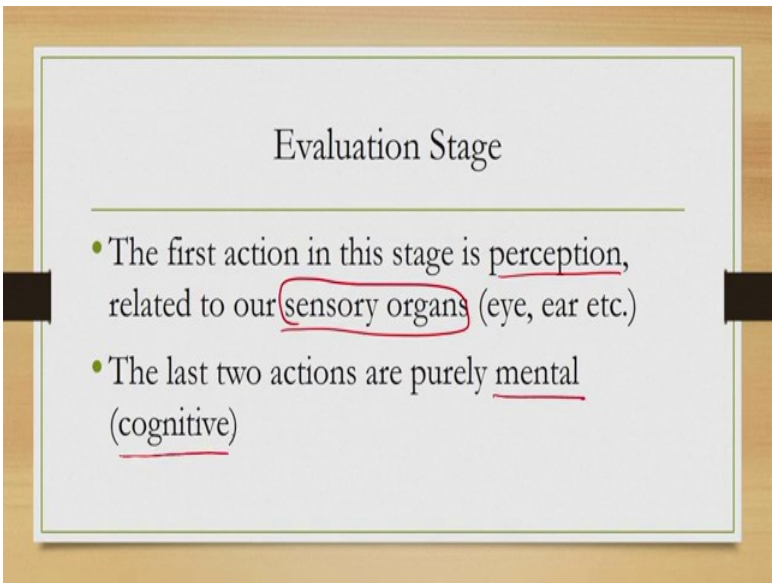
As an example: suppose we perceive that the color of the button changed after we selected it. So, then this is a perceptual input that change of color. Now we might interpret it as the button is selected if there is no color change then we might interpret is as the button is not selected. So, this is the job of the interpretation action.

(Refer Slide Time: 44:49)



Finally, at the end of the evolution stage we evaluate the result of our actions by comparing with the goal state. So, when we have interpreted that the button is selected we compare it with our goal of select the button. And if there is a match that button is selected and our goal was to select the button then of course we do not need to do anything else.

(Refer Slide Time: 45:18)



Like in the execution stage the activities the actions in the evolution stage also are of two types. So, in the last two actions that is interpretation and matching with the gold state

revolution a purely mental cognitive whereas, the first action is perception that is related to our sensory organs. So, first activity is related to sensor; sensory organs and the last two are purely mental or cognitive activities. So, to recollect there are seven actions that will define when we try to understand interactions with an interface.

Now, one thing should be made clear here: although the model talks of interface in general it was primarily used to understand interactions with the GUIs or graphical user interfaces, so, that we should keep in mind. Another thing is the seven actions are divided into two broad groups: the execution stage and the evaluation stage. In the execution stage there are four actions: first we formulate a goal to achieve, second is we formulate our intentions to how to achieve the goal, third is the intention is translated to real world actions that needs to be done to achieve the goal, and finally we actually perform those real world activities.

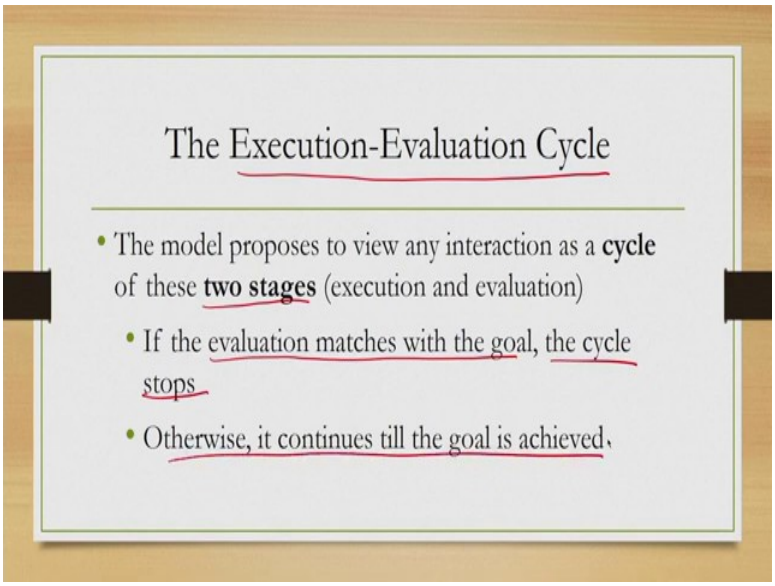
So, among these four the first three setting up of a goal, formulation of intention and translating the intention to real world tasks are purely mental or cognitive activities whereas, the final one involves motor activities; that is used of hand or other motor organs finger. The evaluation stage has three actions: the first one is perceived the state. So, after we executed the real-world tasks in the execution stage the first action that we should do in the evaluation stages perceive; perceive the outcome.

So, that perception may involve visual perception, may involve auditory perception, may involve other modes of perceptions such as touch or maybe a combination of all. The perception is followed by interpretation of the perception. So, whatever we perceived what is the interpretation of that. So, the perception is a sensory activity, the interpretation is a mental activity.

And the last action in the evaluation cycle is evaluation. That is whether I have reached the goal state; whatever I have interpreted whether that is indicative of the goal state. So, this is again another mental activity. So, in the evaluation stage there are two mental or cognitive activities and one sensory activity. In the execution stage there are three mental activities and one motor activity. So, together these seven actions indicate a combination of cognitive motor and sensory activities.

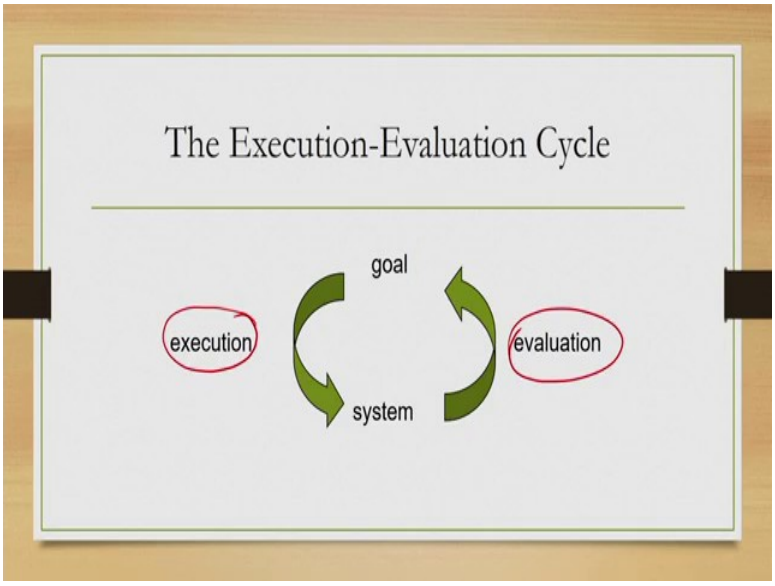
Now there is an important concept that has come out of this model. The concept is known as the execution-evaluation cycle.

(Refer Slide Time: 49:18)



So, when we interact we follow a cycle called the execution evolution cycle. So, the two stages we have discussed; that is execution stage and evolution stage. Now in the evaluation stage if the evaluation matches with the goal the cycle stops, otherwise it continues till the goal is achieved.

(Refer Slide Time: 49:41)

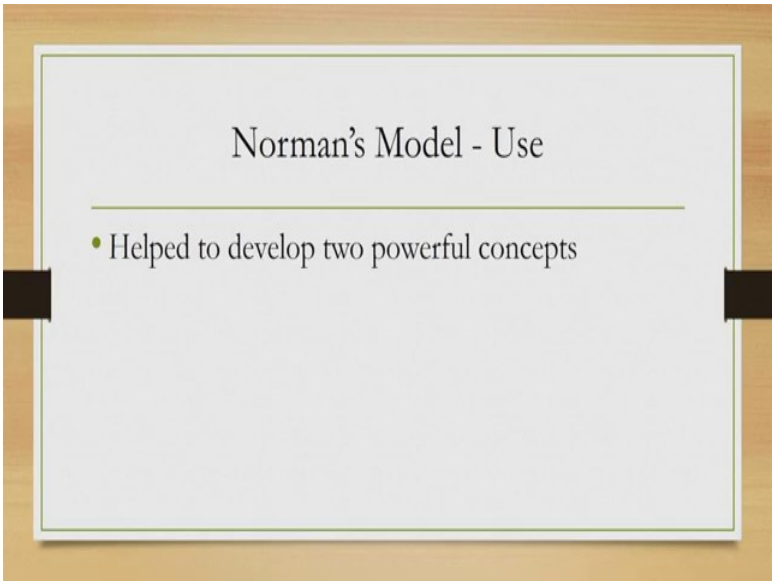


So, we can illustrate the idea with this diagram where, on the one side we have the execution stage on the other side we have the evaluation stage. So, after execution the evaluation stage takes over and there is a comparison between the current state of the

system and the goal state. If there is a match the cycle stops, otherwise again the executions starts and evolution takes over and the cycle continues.

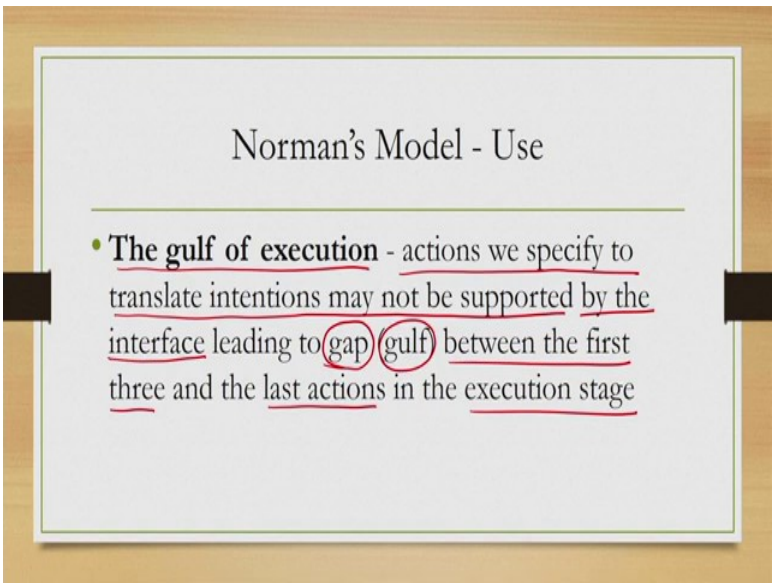
So, this is the execution evolution cycle. This is a nice way of understanding how interactive systems are the how the interaction rather than the interactive system how the interactions take place.

(Refer Slide Time: 50:21)



Now, this concept of cycle in interaction help to develop two powerful concepts.

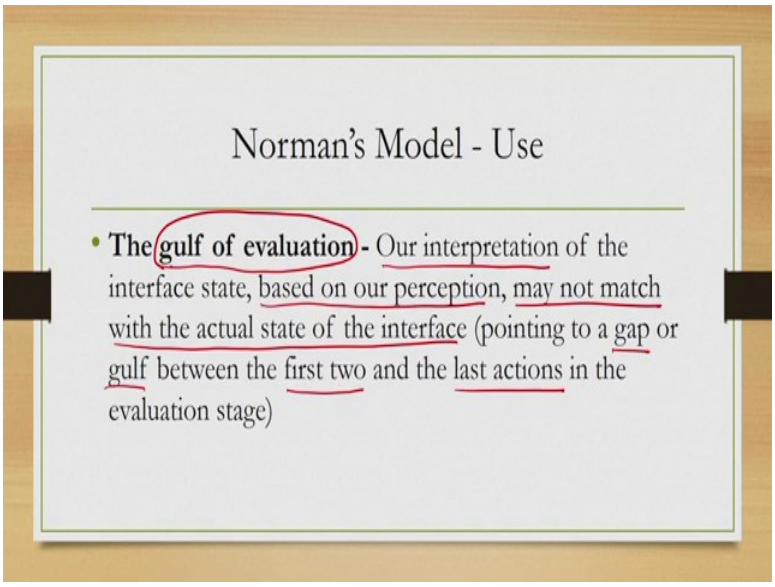
(Refer Slide Time: 50:32)



The gulf of execution: this is the first important concept. Now remember that there are four actions: one action is translate your intention to the real world actions and then execute those actions. Now, if the actions we specify to translate intention may not be supported by the interface, then this leads to a gap or a gulf between the first three and the last actions in the execution stage.

So, when we say that specify the action to be done on the screen by translating your intention to the real-world actions it may happen that the where the system has been designed the translation actually is not working. So, whatever we thought that we should do on the screen those actions are not supported by the interface. So, then it points towards a gap a gulf between different actions in this stage. So, that is known as the gulf of execution.

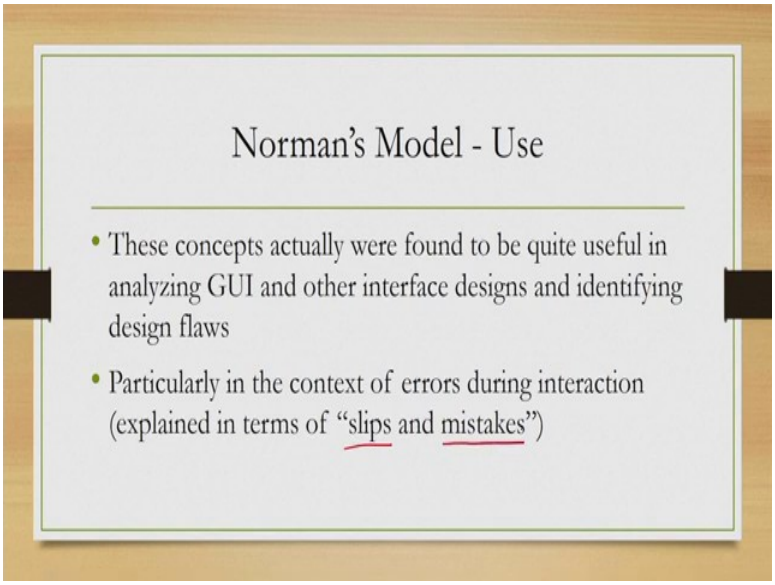
(Refer Slide Time: 51:59)



The other concept is the gulf of evaluation. So, whatever we interpret based on our perception may not match with the actual state of the interface. So, based on the changes that have occurred on the screen we may interpret it in a way which is not actually what happened inside the system. If the design is really bad then that may happen. So, in that case there is a gap or gulf between the first two and the last actions in the evolution stage. So, first two actions are essentially perception and interpretation, and last action is evaluation.

So, based on first two we evaluate come up with a conclusion which may not be true which is due to the gulf in this evolution stage between the actions.

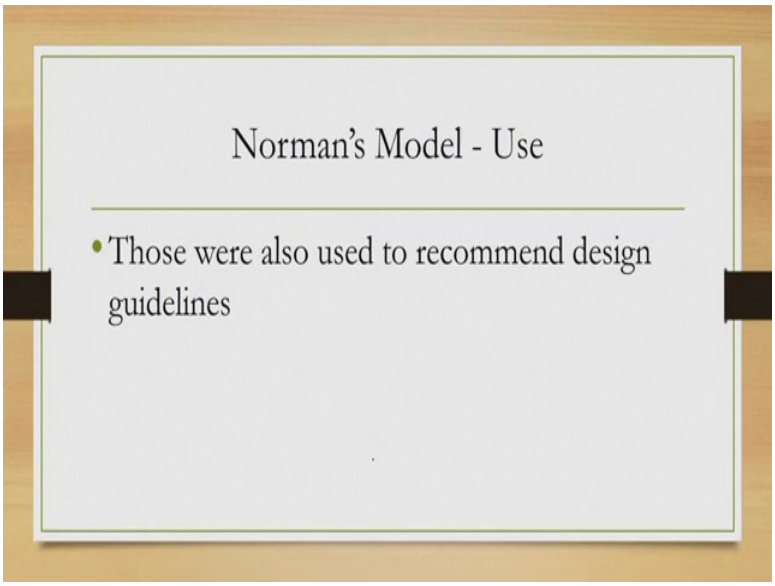
(Refer Slide Time: 52:58)



Now, this two concepts in gulf of execution and gulf of evolution actually helped significantly in coming up with better designs; understanding the design flaws and rectifying those flaws particularly in the context of GUIs. They are very useful in understanding the occurrence of errors and prevention of errors.

So, we will not go into the details of this concept, but just for the sake of completeness would like to mention that two types of errors where define based on these concepts these two gulfs: gulfs of execution and gulfs of evolutions. And depending on the type of errors that are likely to occur for a given design, the design can be modified. So, we can understand using this model of Norman what kind of errors are likely to occur; whether those are slips or mistakes. And accordingly we can make design decisions which are likely to improve the usability of the design.

(Refer Slide Time: 54:18)



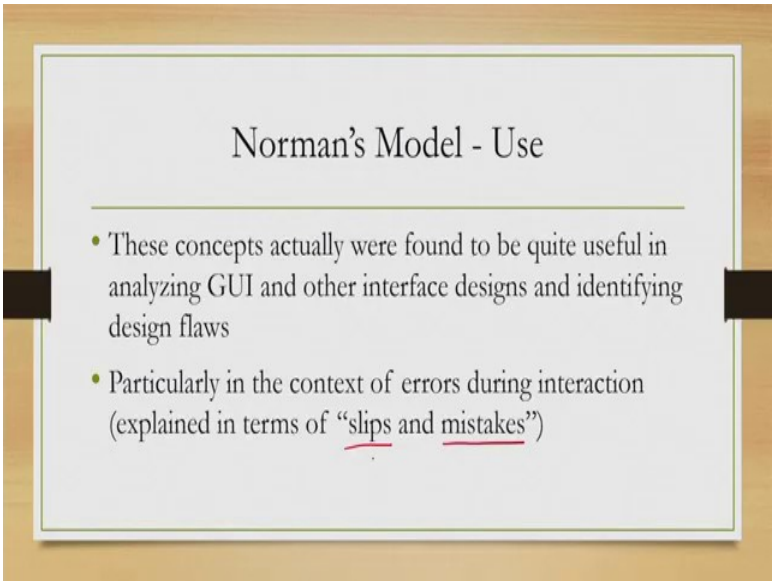
Along with this specific contribution in understanding and preventing the occurrence of errors, the model has also been used extensively to come up with many design guidelines that are extensively used in design of GUIs.

So, let us recollect the Norman's model. It represents the way an interaction takes place from the point of view of the user. So, it is represents interactions as a sequence of seven actions divided into two groups. The two groups are named as execution and evaluation. In the execution group of actions there are four actions namely: formulate a goal, formulate intention, translate intention to real world actions, and execute or actually carry out those activities actions. Three of these are purely mental activity first three the last one is motor activity.

In the evolution stage there are three actions: perceive, interpret, and evaluate. Perceive is a sensory activity, interpret and evaluate are cognitive activities. Now these actions help us in understanding how the interaction takes place and what are likely flaws that may come up if you the design is not done in a proper way. The model in the model of interaction by Norman help to conceive of two important concepts: the gulf of execution and the gulf of evolutions which led to further understanding of the potential design flaws in designs of interactive systems; particularly for GUIs. So that is the advantage of this model.

So, here the model describes interaction, it does not do anything else but from the description we can filter out knowledge that are helpful in coming up with better designs.

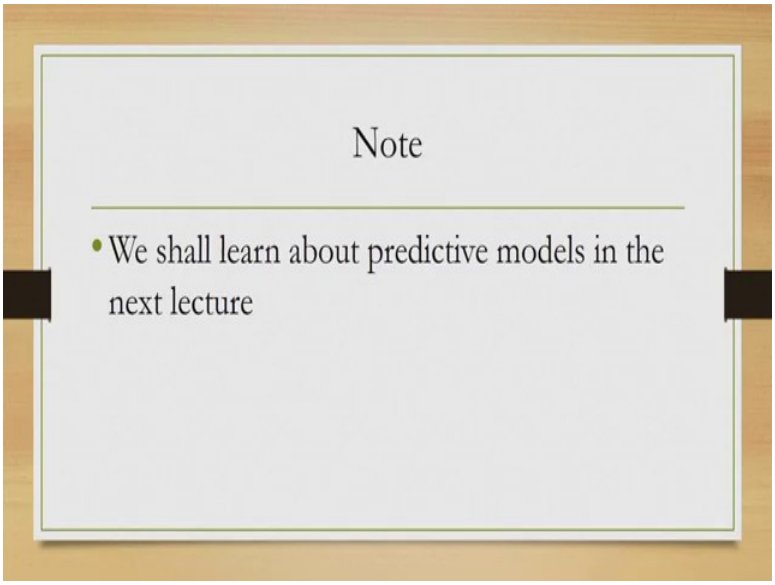
(Refer Slide Time: 57:00)



So, with these three descriptive models we tried to give an idea of what a descriptive model means and what benefit it can bring to the designers, to the developers.

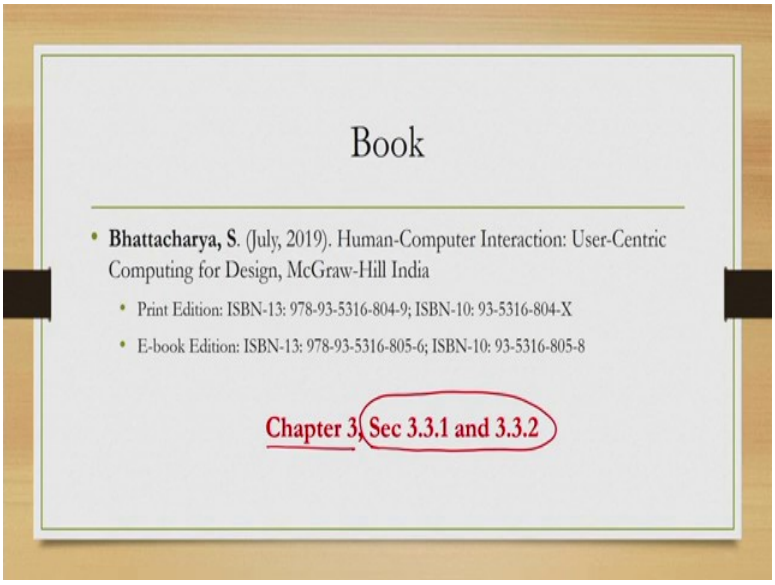
In the next lecture we will discuss about the other class of models that is the predictive models. These models are our main focus in this course, so we will first give an introduction to this class of models in the next lecture.

(Refer Slide Time: 57:32)



In the subsequent lectures we will discuss in details, many of the predictive models that are used in contemporary systems as well as that our classical models no longer used that much.

(Refer Slide Time: 57:51)



The material that we covered today can be found in this book. So, you are advised to consult chapter 3, these two sections: section 3.3.1 and 3.3.2 to find out more details on the topics and related literature that we discussed today.

Thank you and goodbye.