Artificial Intelligence

Prof. Anupam Basu
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

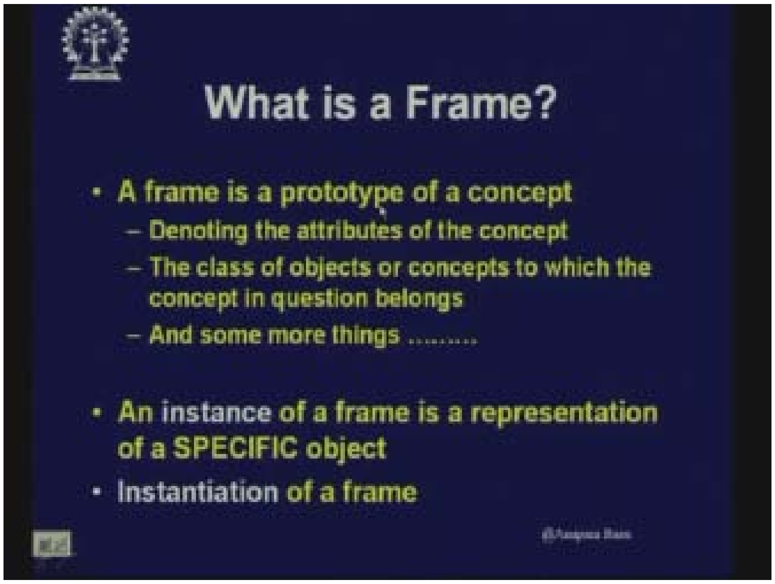
Lecture - 20
Frames

In this lecture today we will start the discussion on a new knowledge representation scheme called frames. Till now we saw two very important knowledge representation schemes; One is rules which is also known as production rules and based on production rules we have seen rule based systems and we have also seen how the inferencing is done in rule based systems. Secondly we have seen semantic nets as a very fundamental knowledge representation method. And in the last two lectures we have also looked into the inferencing method that is adopted in the case of semantic nets.

Today we will be discussing another knowledge representation scheme which is little more systematic but fundamentally it has got many concepts in common with that of semantic nets. And this knowledge representation scheme that we will be discussing today is known as frames. The systems which are based on these frames are known as frame based systems. This system was proposed by a famous computer scientist and artificial intelligence researcher Marvin Minsky and he proposed this scheme in order to represent common sense reasoning.

If you have noted, in case of rules, we have to enumerate all the possibilities to describe a particular state of affairs in its antecedent and if they match then it will fire the rule and the actions corresponding to the consequent will be taken. However this Rule Based System is also known as production rules and one school of thought stated that this is the way human beings represent their knowledge and they take rule based decisions.

(Refer Slide Time 04:00)



What is a Frame?

- A frame is a prototype of a concept
  - Denoting the attributes of the concept
  - The class of objects or concepts to which the concept in question belongs
  - And some more things ....
- An instance of a frame is a representation of a SPECIFIC object
- Instantiation of a frame

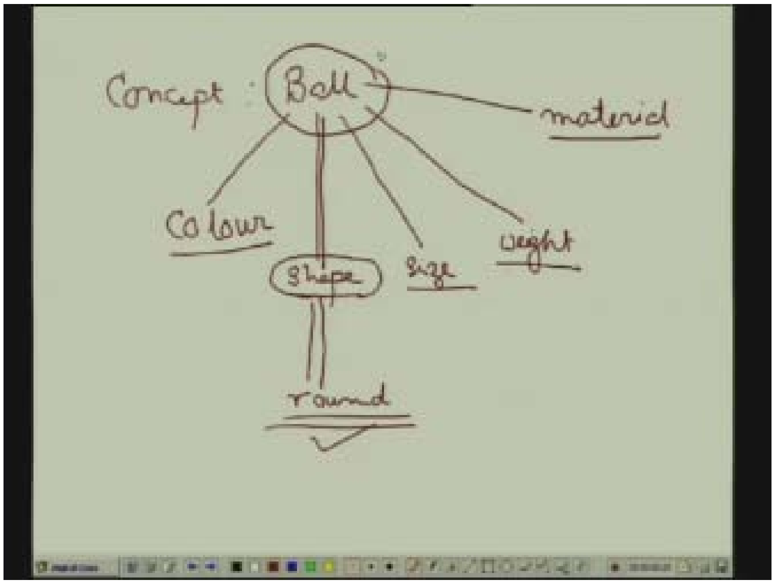
@Anupam Basu

However, Marvin Minsky and other AI decisions by our common sense we do really always fire the rules and match all the antecedents in order to see which rule is applicable in a particular scenario. That was appearing very unnatural to them and so they thought of an alternative knowledge representation mechanism and consequently the frame system was proposed. Now what is a frame? A frame is a prototype of a concept. A very important point over here is prototype.

A prototype it is not the exact description of the concept but it is basically the schema the structure of a particular concept and it denotes the attributes of the concept. Every concept we talk of, for example if we take the concept ball there are so many attributes that we can specify about the ball. May be its color, its shape, its size or the diameter, may be the weight etc and so on and so forth. Now this shape in order to be a ball will be obviously spherical or round. May be the material may vary, whether it is a cricket ball or whether it is a football or whatever. So these are the different attributes associated with this concept ball. While the other attributes can vary one of the attributes like the shape has to be round in order to be qualified as a ball as you see. So that is a particular importance that this attribute shape has.

But right now I am not talking of a particular ball, this is a concept ball, I can have 10 different balls may be 5 footballs, 2 cricket balls etc but those are different instances of the same concept. Similarly we can define a particular concept like bird.

(Refer Slide Time 06:30)



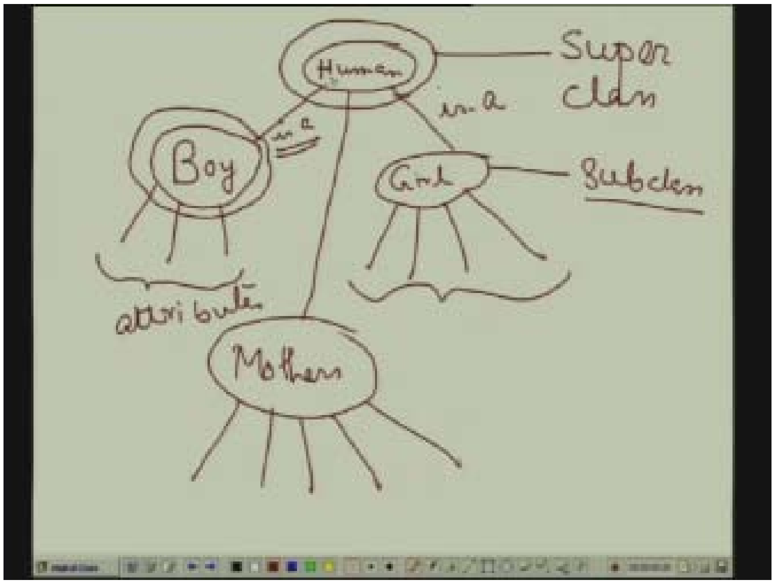
If I define a concept bird then you can start thinking of the different attributes of the bird may be the type or the name of the bird. Now when I say the name of the bird I am not talking of the particular name that I can give to my pet bird but I am talking of a particular species of the bird like may be sparrow. Similarly there can be color of the bird the size of the bird and so on and so forth, now these are the attributes.

Another attribute is can it fly?
Now you can wonder that why all birds fly but that is not the case. For example, ostrich is a bird that does not fly. So, can it fly can have an answer yes or no. the color can be blue, can be yellow, the name can be sparrow, can be parrot so now these rectangles I am drawing here are the attributes of this particular concept and the other ones which I am underlining are the different values. Similarly size can be any real number so these are the values but we are now considering the attributes. So if I want to describe the concept bird then that particular concept has associated with it a number of attributes.

A frame is a prototype of a concept which will describe the attributes of the concept and by that I mean the class of objects or concepts to which the concept in question belongs. Now what we mean by this the class of objects or concepts to which the concept in question belongs. I again take the concept bird or I take the concept boy, now boy is a concept it has got a number of attributes which we have already given some examples of.

Besides this suppose there is a class called human beings. Now boy is a human being and similarly girl is another concept which has got a set of attributes of it but this is again related to another category human beings. Now, all boys are human beings, all girls are human beings, may be all men are human beings so all mothers are human beings and mothers have got their specific attributes. In order to call a lady that she is a mother obviously we look at some attributes. And again mother is a human being. Now what is important over here is that if I think of this human being to be a class then the concept boy is a subclass of this class human. So here is a super class and these are subclasses of the super class. But also denote its class relationship that to which super class it belongs to. So if we just take a set theoretic view of the whole thing then we can also say that human beings are a superset and boy belongs to that set.

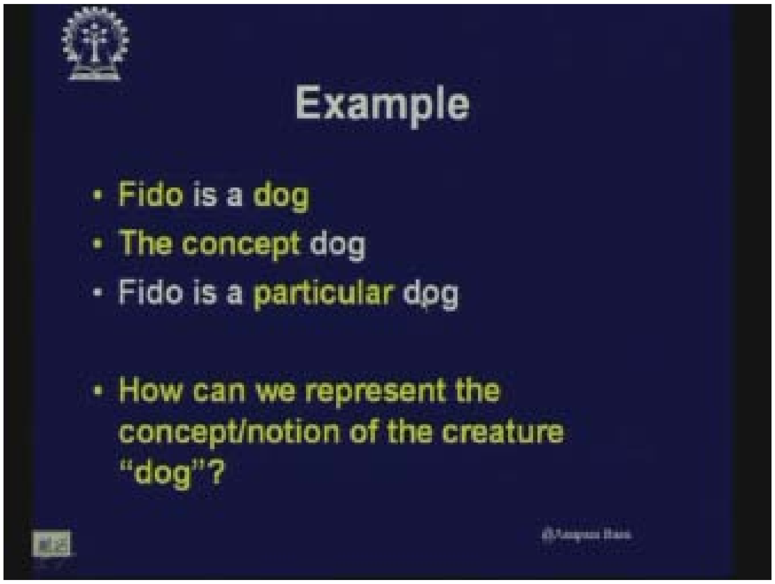
(Refer Slide Time 12:04)



Any boy will belong to that set so this class boy is belonging to this set. Girl is another concept which is a subclass of the class human beings or all the girls are subsets of these class human beings as well. So that is what is meant over here that the class of objects or concepts to which the concept in question belongs. The concept in question in our discussion was a boy or a girl and the class of objects or concepts like human beings to which this concept belongs. The next important thing we like to define is an instance. Now it is very important that we must be able to differentiate between instance of a frame and a frame and instance of a class or the class.

Now I will be using this terms class and instance versus frame and instance of a frame interchangeably because this has got a very close relationship with the object oriented definitions that we are using presently which is very popular now-a-days in different design domains and also in programming languages. And instance of a frame is a representation of a specific object a particular object and that is the instantiation of a frame. Now let us take this example; Fido is a dog, look at these white ones and the yellow ones, Fido is a particular dog and dog is a concept. So we have got the concept dog and Fido is a particular dog. So Fido is an instance of the class dog. Again we can look at it in the way we are drawing.

(Refer Slide Time 15:01)



Example

- Fido is a dog
- The concept dog
- Fido is a particular dog

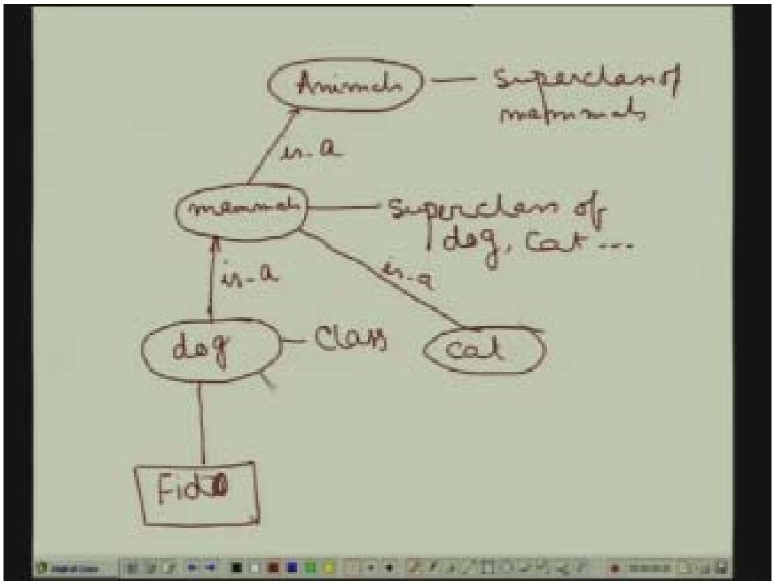
- How can we represent the concept/notion of the creature "dog"?

©Anupam Basu

For example, now I am making a hierarchy animals, now these animals is a class and animals can be mammals or those who are not mammals and dog is a mammal so all these arcs I can say dog is a mammal, all mammals are animals so any mammal is an animal, we very frequently use this is-a relation in the case of semantic nets. So dog is a class because I am not talking of any particular dog, mammals is a super class of dog because dog is a mammal.

Similarly, cat is a mammal so cat is also a mammal so that is another class. Therefore it is a super class of dog, cat etc and animals is a super class of mammals. There can be different animals and some of them are mammals and again among different mammals there may be dogs, cats and other things. There is a sub category which is a dog because there are some attributes which really distinguish a dog from a cat. And if we say Fido is a particular dog, that is I am talking of a particular dog and there may be other dogs which are also which are all dogs may be Jimmy is another dog. I could have made a further classification of these like I can further break this up by doing it like this that dog mammal then comes dog and then among them I can say Spaniards.

(Refer Slide Time 18:01)



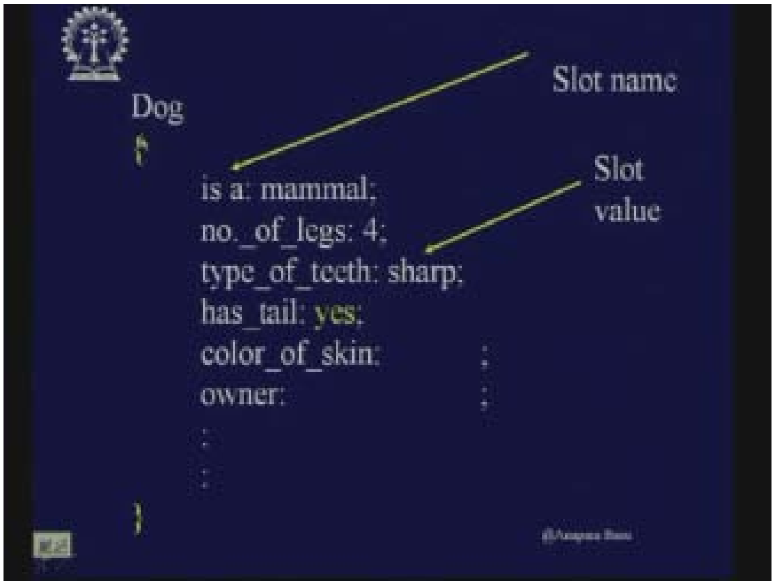
What is a Spaniard?
Spaniard is a class it is a subclass of dog, the Spaniard is a subclass of dog so this is again a class. Now if I say Jimmy is a Spaniard then Jimmy is an instance of relationship instance of a Spaniard. So it is a class and Spaniard is also a dog. Now when we are talking of Jimmy then we are talking of a particular element of this class. So the same thing we can draw in the form of a set like the largest set is animal. In this there is a subset which is mammal in which there are dogs or there may be cats and there can be Spaniards and all those.

Now, when I talk of Jimmy and I talk of Fido they are different elements of this set. So, in terms of the set theoretic notation you can understand that each of these elements is particular instances of a class and that class or the particular set. Now anything that belongs to this set obviously belongs to this set. Or if I say dog and Jimmy is an instance of dog and dog is a mammal now this is an instance relationship and this is a is-a relationship. So Jimmy is an instance of dog and dog being a mammal Jimmy is also a mammal so this is a transitive relation. This is the difference between instance of a class and the class itself.

Now by frame will be denoting the class and by instances of the frame will denote the instances. Now let us see how we can represent the concept the notion of the feature dog.

The description of a frame is very much like the structure data type that we use in the language C. A frame consists of a number of slots. Each of these attributes dog is a concept which has got a number of attributes. The attributes are: number of legs, type of teeth, has tail, color of skin, owner etc. Now these slots can have values and the values can be of different types. So I represent the frame or the concept, the frame representing the concept dog as a set of attributes it reads dog is a mammal that means it is saying that the concept dog is connected to the concept mammal or the class dog is connected to the class mammal through the is-a relationship besides these attributes like number of legs, type of teeth, has tail, color of skin, owner etc.

(Refer Slide Time 23:02)



```
Dog
{
    is a: mammal;
    no._of_legs: 4;
    type_of_teeth: sharp;
    has_tail: yes;
    color_of_skin:          ;
    owner:                  ;
    :
    :
    :
}
```

These are the slot names and these slots can have values which are called the slot values. You may also note that the different values these slots can have are not all of the same type. for example, dog is a mammal so this is a string a class name, number of legs is an integer, type of teeth is again a string, has teeth is a Boolean yes or no, color of skin will again be string, owner will be string and there can be owner of the dog which can be the string. When we are describing dog we are saying the frame dog its slot has tail has a value yes which means dog has tail. But obviously there can be dogs which have got their tails cut we often see such dogs in different places.

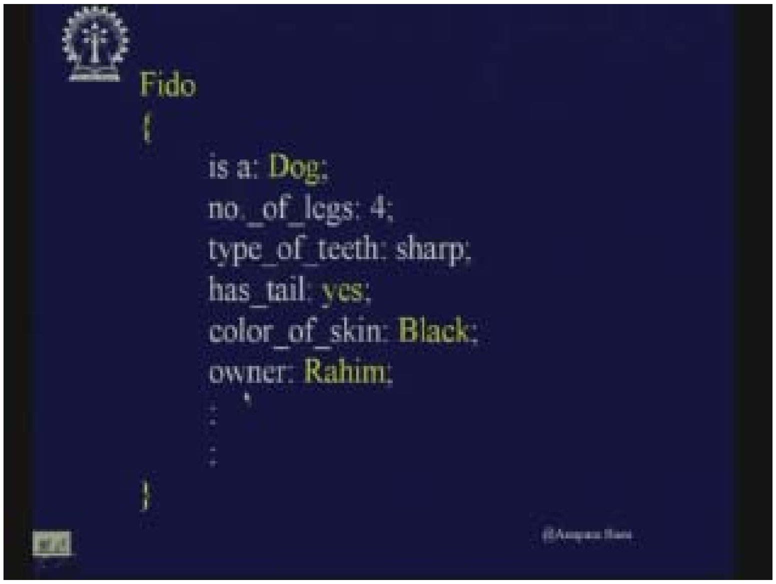
Then is it the case that all dogs have tails?

Here what we are saying is this value is a special type of value which is known as a default value that is unless we know that the tail of the dog has been chopped off then using our common sense we can assume that if it is a dog then obviously it has a tail. So through this description I really build up the concept of the dog and I need not fire ten rules in order to infer that dog has a tail. The notion of frame is that it is felt by some people that when we learn different concepts along with that we also learn the attributes of the concept and the different values of those attributes. So unless we know something different we have seen so many dogs from our childhood and we have learnt from our basic science book that dog is a mammal and we have seen and learnt that the number of legs is four so as soon as the word dog is uttered all these things are activated in our mind.

Now I could not put a value over here because dogs may have owners except for street dogs might be but for the concept dog there is no particular owner that is tied to it. For a particular dog it might have some owner. We again describe Fido.

Now what is Fido?
Fido is a particular dog therefore Fido is an instance of the concept dog. Let us see how the slot values have changed. Fido is a dog so I could have written here also Fido is instance of dog that would have been more correct. Now number of legs is four because Fido is a dog so it has got number of legs four, type of teeth sharp, has tail yes, color of skin black, owner here, owner I can put the value Rahim and even earlier also the color of skin was black because I cannot attach any particular value to the color of the skin of the concept dog. Now when I describe this that is automatic so it is a constant and dog is a mammal and dog had attributes like type of teeth and the value was sharp and the number of legs four, color of skin I did not put any value, owner I did not put any value.

(Refer Slide Time 28:02)



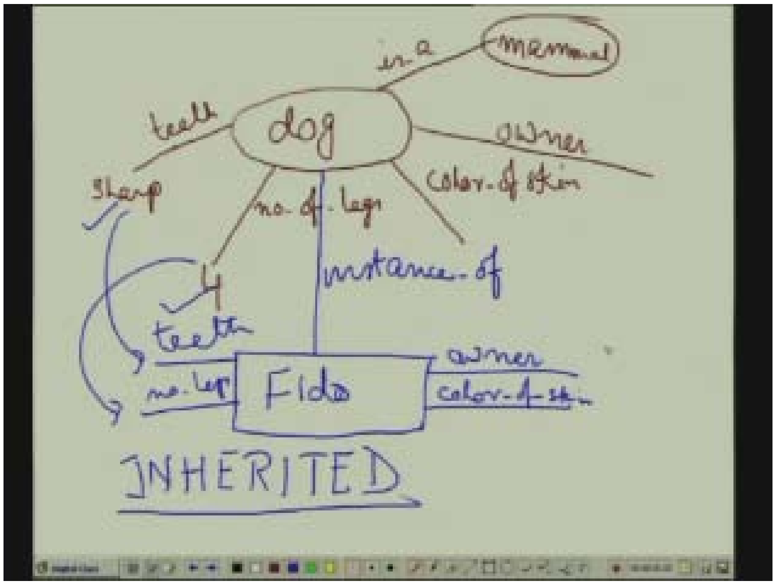
```
Fido
{
        is a: Dog;
        no._of_legs: 4;
        type_of_teeth: sharp;
        has_tail: yes;
        color_of_skin: Black;
        owner: Rahim;
        :
        :
}
```

@Anupam Basu

Now I select another color and say that Fido is an instance of dog. Now when I say this automatically I specify many things without specifying them with Fido. Here I do not need to specify that Fido has four legs, Fido’s teeth is sharp because since Fido is an instance of dog some of these attributes which are specified will automatically be inherited in the instance. You are familiar with this term inheritance through our discussion on semantic nets. These attributes like teeth, number of legs are automatically inherited from the class definition. Those attributes like owner which is not specified in the parent class or the color of skin which was not specified in the parent class can be specified here. I can say it is black and owner may be Rahim.

Now this is one very important advantage of frames and this advantage is derived from the inheritance we have just now shown. And a very important advantage for any knowledge representation is the economy of representation.

(Refer Slide Time 30:57)



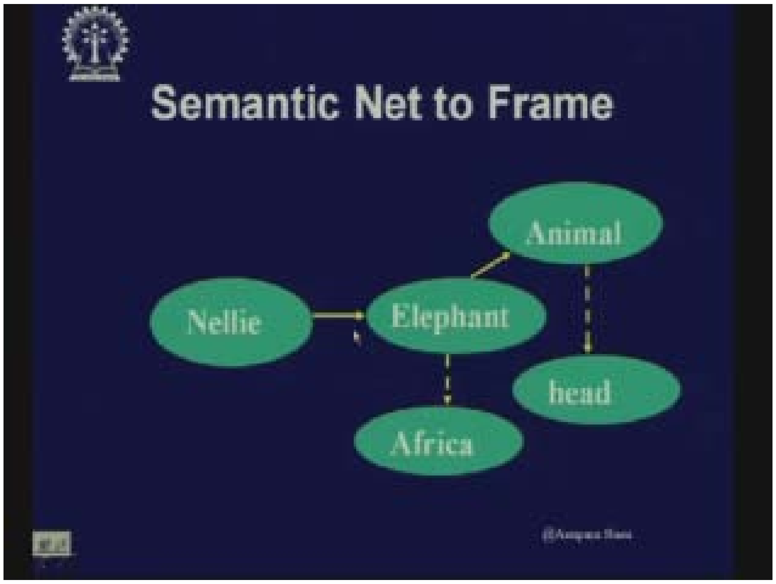
Even if I can infer many things I need not explicitly specify it all the time hence obviously it is an advantage. The economy of representation is that I am specifying less but I can infer more and this seems to be always attractive. All these need not be specified but they are inherited automatically and I can build up a C structure like this with these values. Any slot which does not have a value or we say that the slot is not instantiated it means that this slot has not got a particular value as yet we will infer that to be unknown. Now we come to a common notion which we have seen earlier that is the concept of object attribute value.

While discussing Rule Based System and semantic nets we mentioned that we can represent many facts of the real life using this object attribute value triple. So an instance of a frame is denoting an object like frames like a dog. The slots are denoting the attributes of the object and the slots can be of different types as we have seen and the slots are instantiated with values. So, if we go up, if I just make a statement Fido is a dog and that is automatic here it is represented here because the object name is Fido, attribute name is-a and the value is dog. So you can also see how equivalently it translates into a proposition. If I take this object or instance and take this attribute and take this value what would be the equivalent proposition? If I write in the well known propositional logic we will say number of legs of Fido is four and we have seen that the slots can be of different types and the slots are instantiated with values.

Now let us look at the correlation between semantic nets and frames. Here is a semantic net we showed earlier in the last lecture. In semantic net we draw like this everything is like oval. Nellie is a elephant, elephant is a animal, in semantic nets we draw everything in the same way, I have intentionally not written these attributes because I hope you will be able to say what this attribute is, animal and head how are they related? I can say animal has part head, elephant and Africa how are they related? They are related with elephant found at or lives in Africa. And Nellie is an elephant, elephant is an animal.

Suppose I label these edges and if this information is to be translated to frames then what would be the class and what would be the instance? Based on this we will be now drawing the frame or we will show just what the slots are. Now if I look at elephant it is a class so according to this semantic these are class relationships and there are other attributes and also elephant found in Africa, elephant has part head.

(Refer Slide Time 36:20)



Semantic Net to Frame

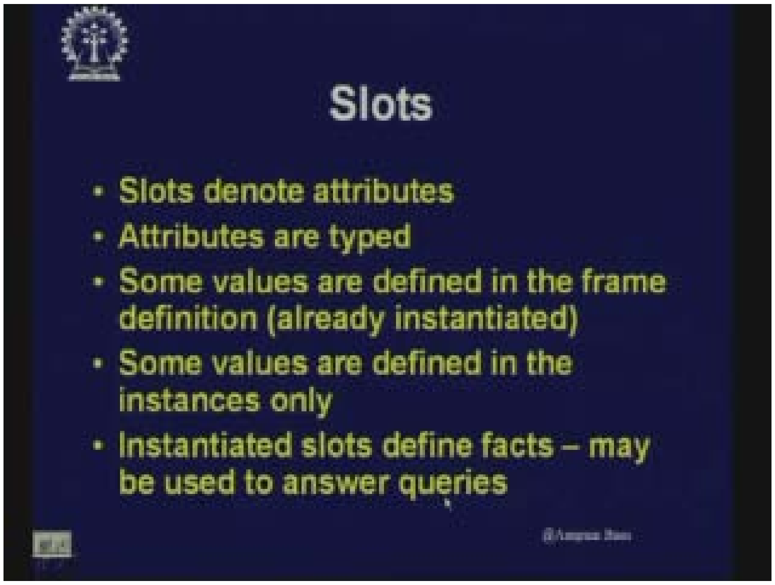
@Anupam Basu

I have intentionally used two different colors so here you see the class is denoted by the frame name, the relations of the semantic net are actually the slot names and we have got the slot values. So if we have Nellie which is an instance we will say it is an instance of elephant and other attributes. So if we go back here the concepts are translating to either the frame or the instance of a frame Nellie is an instance and the relations are translating to the slot names and we are putting in the slot values. Now this distinction was not there in the case of semantic nets.

What are slots?
Slots denote attributes, attributes are typed because there can be different types, some values are defined in the frame definition and already instantiated like we have seen in the case of Fido and the dog that some are already known, some values are defined in the instances only, instantiated slots are defining facts or propositions which we have seen and this may be used to answer queries. If I look at this instance then I can answer the query if the question is put does Fido has tail? I will come to Fido I will go to the has tail slot and I will look at the value if there is no value I will say unknown but if there is a

value I will say yes. Who is the owner of Fido? So I look at the object, what is the object? It is Fido and what is the attribute that I am questioning? It is the owner so I come to this owner slot and what is the value of that? It is Rahim and therefore my answer to the question is that the owner of Fido is Rahim. So this instances or this representation helps us to answer queries.

(Refer Slide Time 40:02)



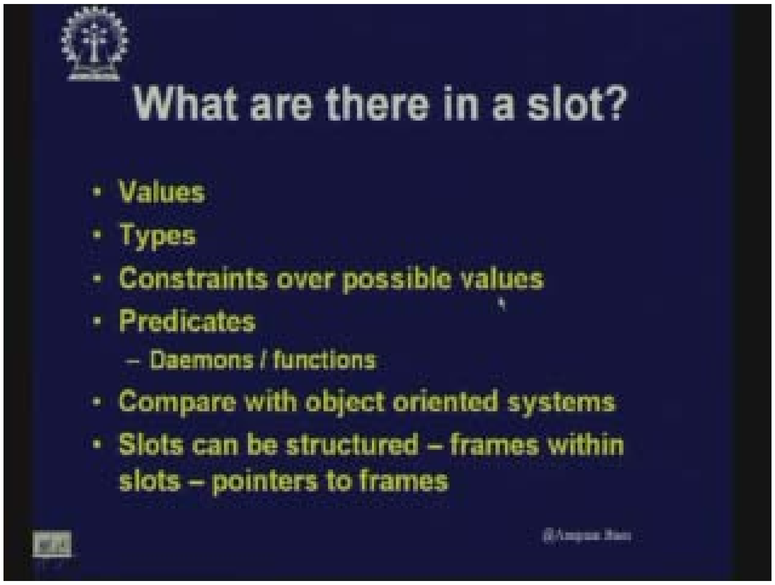
Slots

- Slots denote attributes
- Attributes are typed
- Some values are defined in the frame definition (already instantiated)
- Some values are defined in the instances only
- Instantiated slots define facts – may be used to answer queries

Next ask the question what are there in a slot? Obviously in a slot there will be values, there will be types which will say what type here I am introducing two new things which you do not know that is constraints over possible values. I cannot allow any value not only type there may be a range beyond which I cannot put in the value. Also we put in some predicates now these predicates are also known as daemons, they are actually functions.

If you compare it with object oriented systems, we basically know that there are concepts of classes and objects of a class. Now if I make a mapping then I can say in an object oriented system and side by side I look at the frame based system so I have got classes in object oriented systems and they are mapping to frames there is a mapping. I have got objects in object oriented system and that is mapping to instances in the case of object oriented systems.

(Refer Slide Time 41:29)



What are there in a slot?

- Values
- Types
- Constraints over possible values
- Predicates
  - Daemons / functions
- Compare with object oriented systems
- Slots can be structured – frames within slots – pointers to frames

@Anupam Basu

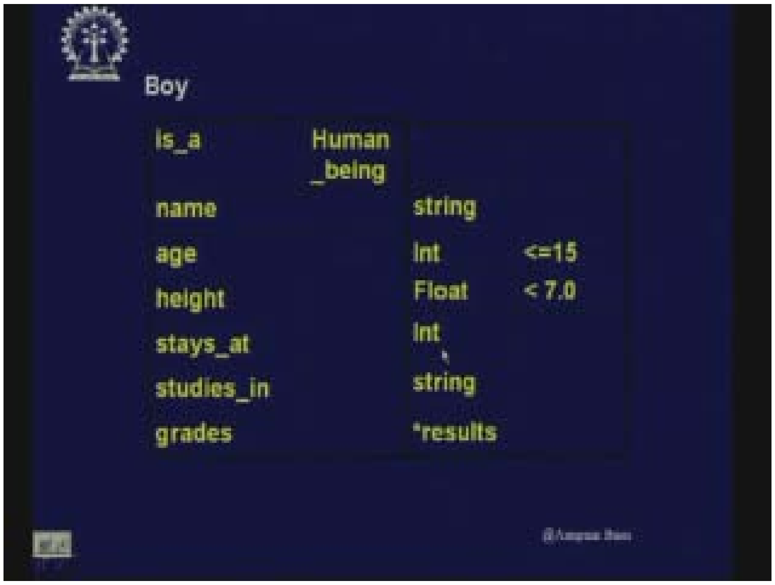
In object oriented systems the objects communicate using messages and when an object receives a message then it fires or activates some methods and these methods are equivalent to these daemons or these are also used as predicates. So there is a one to one correspondence between these.

Another point is; slots can be structured or there can be frames within a particular slot and we can use pointers to another frame. Let us look at this example; Boy is a human being so is-a is a slot name and human being is a value. Name is a string, now here for the boy the name part is not instantiated but the type is specified that whatever values it may have should be a string, age will be an integer. Here in order that I can really call somebody a boy the age must be less than or equal to 15, height type is float and height should be less than 7 otherwise that is a spurious or wrong value that is coming, studies will be string here so there are no constraints over here.

Type is a pointer to result so result is another frame that means if I need grades or the c g p then for that boy there is a grade and the grade can be computed from the results whatever marks he has got based on that I can find out whether he gets a or b or c so it is a pointer and pointer to another frame which is known as results and that is also possible and that is what is meant by this last line slots can be structured that means within a slot I can have a frame. The next thing is in addition to these types I can put in predicates.

What are these predicates?
There are two types of predicates in general namely if added and if needed.

(Refer Slide Time 44:37)



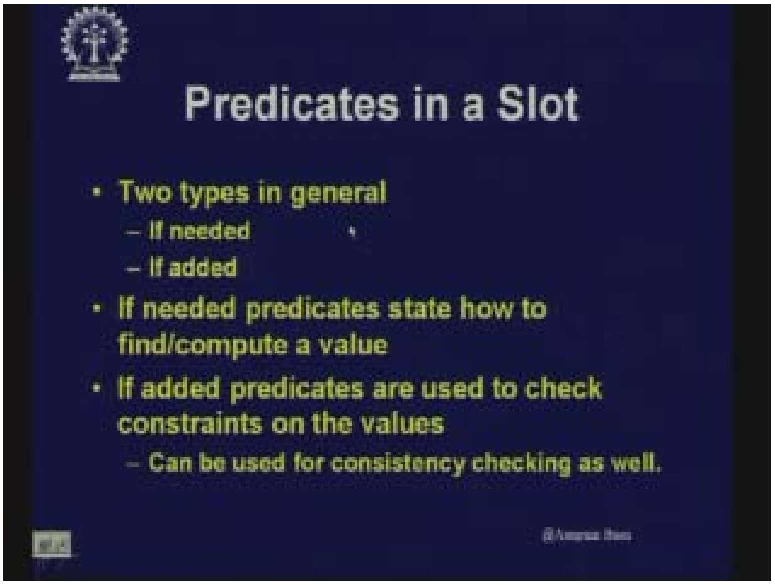
Boy

is_a	Human _being		
name		string	
age		int	<=15
height		Float	< 7.0
stays_at		int	
studies_in		string	
grades		*results	

If needed predicate state how to find compute a value. So if you need a particular value to be computed for some inference that particular predicate will tell you how you can compute it. And if added predicates are used to check the constraints on the values that can be used for consistency checking also. So if I need grades here is along with the slot I have put in a predicate. If needed that if at any particular point of time I need this for inference I will fire this function I will activate this function compute c g p a may be with parameter boy and results because I will go to the result frame and will compute the c g p a. Suppose from some other inferencing I get the value of height and the height is 7.5 feet that will violate the constraint that has been put in because if some value is put in I want to check whether it is the correct value or not so I have got a constraint here.

So, if the height is added then I have to check this constraint. Now let us look at Tom. Tom is again an instance of boy, name is Tom smith, age is 12 so all these things are instantiated, height is 5, stays at 15 park avenue, studies in bright academy, grades now grades I will say if needed compute the c g p a of Tom from results of Tom. Let us quickly recapitulate what we have said.

(Refer Slide Time 46:01)



Predicates in a Slot

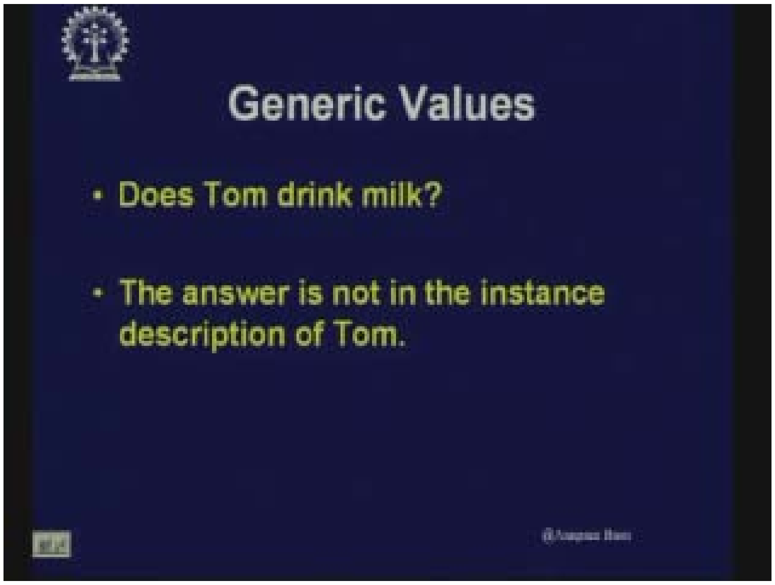
- Two types in general
  - If needed
  - If added
- If needed predicates state how to find/compute a value
- If added predicates are used to check constraints on the values
  - Can be used for consistency checking as well.

@Anupam Basu

In a slot we have got the slot names, along with every slot we have got its type, we have got some constraints, the slots can be instantiated with values, we can have constraints with the slots, also we can have predicates with the slots, two types of predicates if needed and if added. Let us see how if needed predicates are used for inferencing. Now, before going to the inferencing we will quickly talk about two types of values. There are some things called generic values.

Generic values like Tom drink milk; can we get the answer from this description of Tom? No, there is no slot of Tom so how we carry on inferencing. I have got a question does Tom drink milk? I do not find any attribute about drinking over here but I try instance of, so what would I do? I have got an instance Tom, now here I am pointing in the inference part also now Tom has got so many attributes but drinking milk is not there. But I know Tom is an instance of boy now I can go to boy and see if there are some attributes talking about some drinking. Unfortunately in the description of boy we showed there was no such attribute so we go up and we find that boy is a baby human being and human beings drink milk because they are mammals.

(Refer Slide Time 48:44)



Generic Values

- Does Tom drink milk?
- The answer is not in the instance description of Tom.

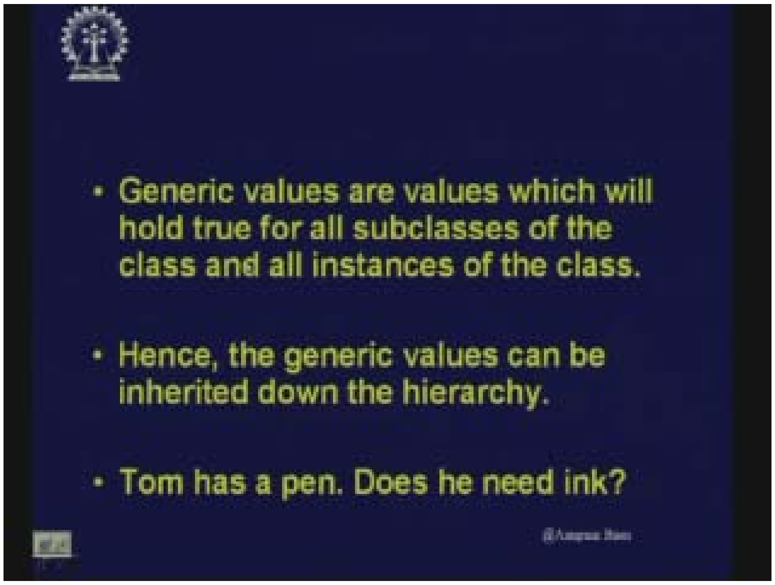
@Anupam Basu

Now, all human beings drink milk because they are mammals. So this is a generic property of this class. Therefore any property that is holding for all elements of this class is known as a generic property. So although we do not know whether Tom drinks milk and since Tom is an instance of a boy the inference machine will try whether boy drinks milk, the answer is not there so I go up here and see that boy is a human being and human being drinks milk. Now this information is true for all human beings so that is a generic value.

What is the generic value?
Tom is an instance of a boy and also boy is a human being so in the description of human being we may have human beings drink milk and since all human beings drink milk the answer can therefore be inherited and inferred through inheritance. We have also seen this for semantic nets. Generic values are the values which will hold through for all sub classes. Similarly you can try, Tom has a pen does he need ink? Suppose Tom has a pen does he need ink? Now how do you answer that? In the description of Tom it is not there that Tom needs ink but it is just said Tom has a pen. Therefore, we can go to the class 'pen' and find out that pen needs ink. Therefore we inherit the answer down the line and we infer that Tom needs ink.

Besides generic value there is a very important type; does Tom write? Again the answer is not in the description of Tom. Whatever we have described for Tom it was not there that through which hand he writes, it was not there so what can we do? We will suppose in the description of boy we have another slot writes with right hand. Now I have put a star here to show that this value is a different type of value what is that? In general that is unless anything is known to the contrary.

(Refer Slide Time 52:27)



- Generic values are values which will hold true for all subclasses of the class and all instances of the class.
- Hence, the generic values can be inherited down the hierarchy.
- Tom has a pen. Does he need ink?

@Anupam Basu

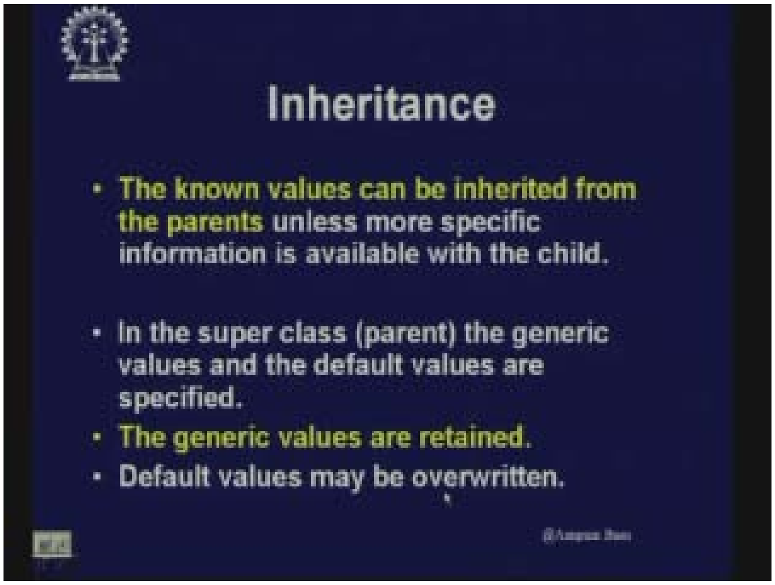
I will assume that a boy writes with right hand but it may not be true so these type of values which we assume since we are not told that Tom is left handed we will assume that Tom writes with right hand. This is known as the default value that we assume in the absence of any more information. Now again you can see that the default is with right hand. There is a difference between generic values and default values, if while describing Tom I get some additional information that Tom is left handed obviously the default value will be overwritten. However, the generic values cannot be overwritten if it is overwritten the generic values are tightly connected to the class, if the generic value is changed the class definition changes. Inheritance is a very powerful mechanism and we have seen inheritance so many times in semantic nets. The known values can be inherited from the parents unless more specific information is available with the child.

In the super class or the parent the generic values and the default values are specified. The generic values are retained and the default values may be overwritten as we have already seen. Multiple inheritance is one issue which is little difficult to solve. That is a class or instance may be a child of multiple parents. Now what do we need? For example I define a class teacher politician. A teacher politician is a teacher as well as a politician. Now for a teacher the top priority is qualification whereas for a politician the top priority is the vote bag. Now I say Mr. X is a teacher politician.

What is the top priority of Mr. X?
It is very difficult to say so we have to take some policy and depending on the context we may decide which property which parent's property we will inherit. similarly you can see another example; scooter I can say automobile and two wheeler and two wheelers have number of wheels two, automobile is four so if I say scooter then how many wheels does scooter have? But this is a wrong organization, this problem should not have occurred, the correct organization should have been this: two wheeler is a automobile, scooter is a two wheeler in that case when I try to answer the query how many wheels does the

scooter have I will go to the top the first one and try to inherit this. So what is the inference mechanism in frames? The inference mechanism is simple, look for a candidate frame or instances, look for the availability of values if the value is available then resolve else go to the preceding level and so on if solved at a higher level try to have a more specific answer.

(Refer Slide Time 55:52)



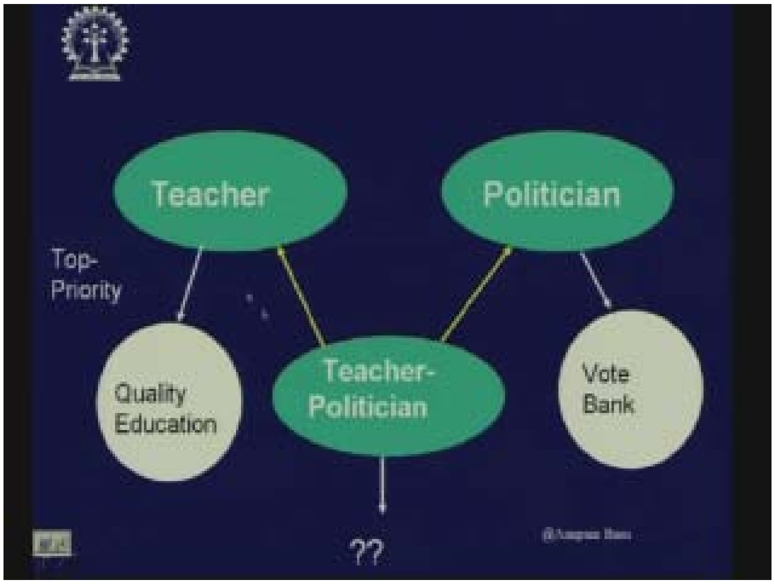
Inheritance

- The known values can be inherited from the parents unless more specific information is available with the child.
- In the super class (parent) the generic values and the default values are specified.
- The generic values are retained.
- Default values may be overwritten.

@Anupam Basu

Now I will just illustrate this with this, suppose how many wheels does scooter have?

(Refer Slide Time 56:06)



Teacher

Politician

Top-Priority

Quality Education

Teacher-Politician

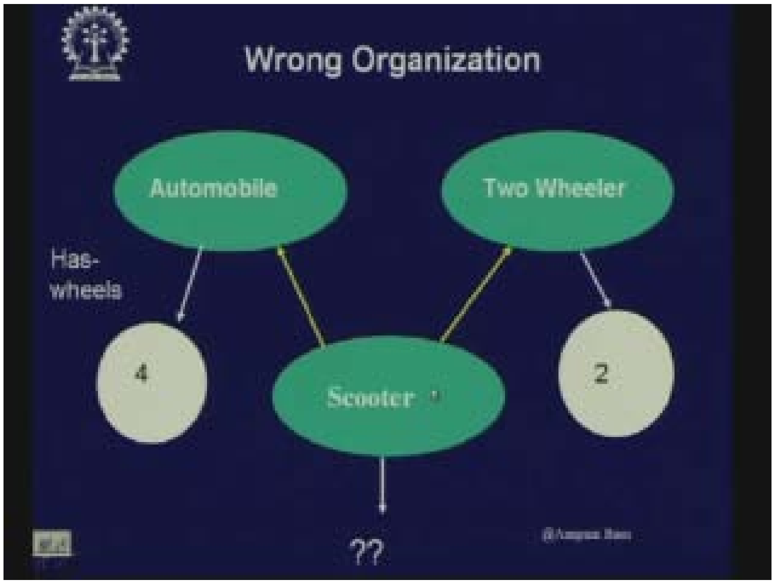
Vote Bank

??

@Anupam Basu

I do not find that value as an attribute but since I do not specify everything everywhere I go up one and find that scooter is a two wheeler and it has got two wheels therefore I infer by inheritance that scooter has two wheels. If this was not specified I would have gone one level higher and would have taken the higher level information inferencing. However this is just to give you the overview of frames. There are some subtleties in frame based system.

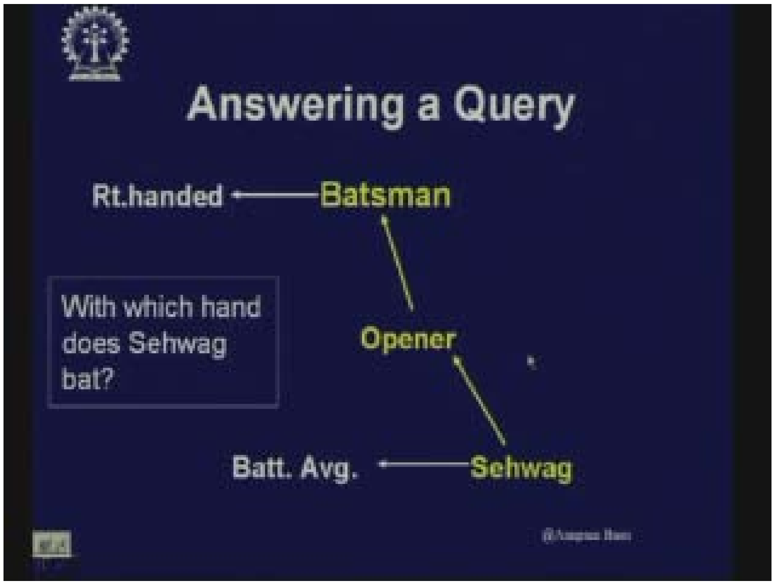
(Refer Slide Time 56:43)



Wrong Organization

These are very important in answering queries for example say with which hand does Virendar Sehwag bat? Now Sehwag has been described with a batting average value that attribute is here. Now Sehwag is the opener but here I do not have anything saying with which hand he bats, I go up and opener is a batsman and in batsmen I can say there is a default value specified that is right handed and I can specify that Sehwag is right handed. In the same way frames can be used for classification of objects and there are many other applications.

(Refer Slide Time 59:04)



Answering a Query

Rt.handed

Batsman

With which hand does Sehwag bat?

Opener

Batt. Avg.

Sehwag

But for this lecture I think this is sufficient and I will encourage you to describe different scenarios that cars automobiles or objects that you see around and try to make frame based representation and also utilize the examples that has been given in the frames.