Approaches to knowledge representation:

There are mainly four approaches to knowledge representation, which are given below:

1. Simple relational knowledge
2. Inheritable knowledge
3. Inferential knowledge
4. Procedural knowledge

1. Simple relational knowledge:

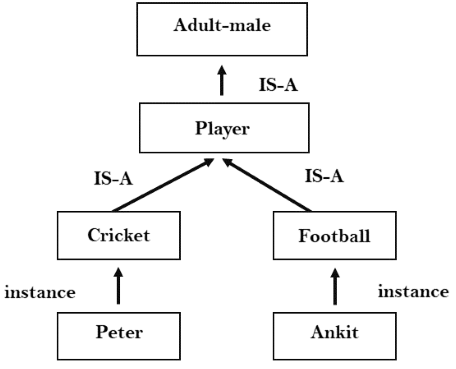
* It is the simplest way of storing facts which uses the relational method, and each fact about a set of the object is set out systematically in columns.
* This approach of knowledge representation is famous in database systems where the relationship between different entities is represented.
* This approach has little opportunity for inference.

**Example: The following is the simple relational knowledge representation.**

|  |  |  |
| --- | --- | --- |
| **Player** | **Weight** | **Age** |
| Player1 | 65 | 23 |
| Player2 | 58 | 18 |
| Player3 | 75 | 24 |

2. Inheritable knowledge:

* In the inheritable knowledge approach, all data must be stored into a hierarchy of classes.
* All classes should be arranged in a generalized form or a hierarchal manner.
* In this approach, we apply inheritance property.
* Elements inherit values from other members of a class.
* This approach contains inheritable knowledge which shows a relation between instance and class, and it is called instance relation.
* Every individual frame can represent the collection of attributes and its value.
* In this approach, objects and values are represented in Boxed nodes.
* We use Arrows which point from objects to their values.
* **Example:**



3. Inferential knowledge:

* Inferential knowledge approach represents knowledge in the form of formal logics.
* This approach can be used to derive more facts.
* It guaranteed correctness.
* **Example:** Let's suppose there are two statements:
  1. Marcus is a man
  2. All men are mortal  
     Then it can represent as;  
       
     **man(Marcus)  
     ∀x = man (x) ----------> mortal (x)s**

4. Procedural knowledge:

* Procedural knowledge approach uses small programs and codes which describes how to do specific things, and how to proceed.
* In this approach, one important rule is used which is **If-Then rule**.
* In this knowledge, we can use various coding languages such as **LISP language** and **Prolog language**.
* We can easily represent heuristic or domain-specific knowledge using this approach.
* But it is not necessary that we can represent all cases in this approach.

Requirements(OR PROPERTIES) for knowledge Representation system:

A good knowledge representation system must possess the following properties.

1. **Representational Accuracy:**  
   KR system should have the ability to represent all kind of required knowledge.
2. **Inferential Adequacy:**  
   KR system should have ability to manipulate the representational structures to produce new knowledge corresponding to existing structure.

**EX:** if it is raining john never goes out + it is raining today

We can conclude that today john will not go out

1. **Inferential Efficiency:**  
   The ability to direct the inferential mechanisms into the most productivedirections by storing appropriate guides.
2. **Acquisitional efficiency-** The ability to acquire the new knowledge easily using automatic methods.

Knowledge Representation using Semantic Network

* + Semantic networks (semantic net) are a commonly used representation in AI. In Semantic net, we can represent our knowledge in the form of graphical networks.
  + A semantic network is a graph consisting of nodes that are connected by edges.
  + Nodes represent objects, and the links between nodes represent relationships between those objects. The links are usually labelled to indicate the nature of the relationship.

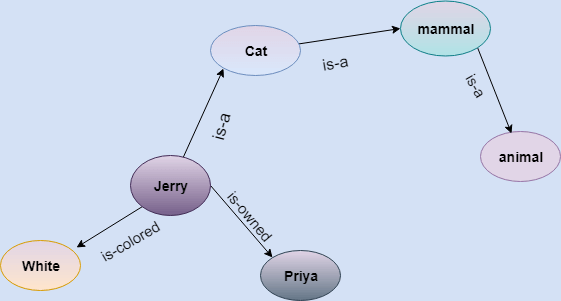
This representation consists of mainly two types of relations:

* 1. IS-A relation (Inheritance)
  2. Kind-of-relation

**Example:** Following are some statements which we need to represent in the form of nodes and links.

Statements:

1. Jerry is a cat.
2. Jerry is a mammal
3. Jerry is owned by Priya.
4. Jerry is brown colored.
5. All Mammals are animal.



In the above diagram, we have represented the different type of knowledge in the form of nodes and edges. Each object is connected with another object by some relation.

Drawbacks in Semantic representation:

* 1. Semantic networks take more computational time at runtime as we need to traverse the complete network tree to answer some questions. It might be possible in the worst case scenario that after traversing the entire tree, we find that the solution does not exist in this network.
  2. Semantic networks try to model **human-like memory** (Which has 1015 neurons and links) to store the information, but in practice, it is not possible to build such a vast semantic network.
  3. These types of representations are inadequate as they do not have any equivalent quantifier, e.g., for all, for some, none, etc.
  4. Semantic networks do not have any standard definition for the link names.

Advantages of Semantic network:

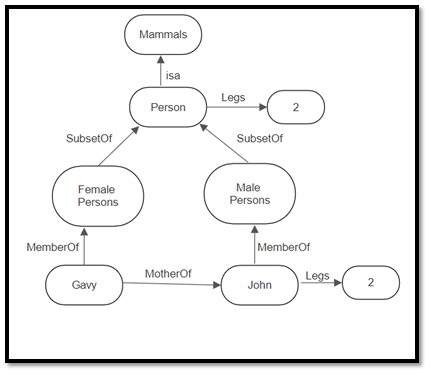
1. Semantic networks are a natural representation of knowledge.
2. Semantic networks convey meaning in a transparent manner.
3. These networks are simple and easily understandable.

# Inheritance in Semantic Net

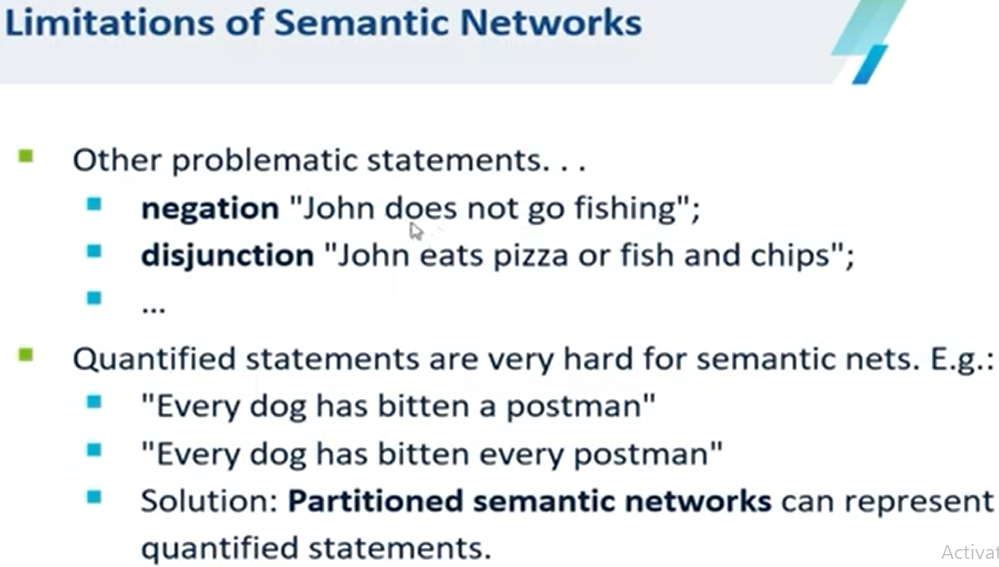
* Inheritance allows us to specify properties of a super class and then to define a sub class, which inherits the properties of the super class.

In our example, mammals are the super class of PERSON. Female Persons & Male Persons are sub classes of Person Class

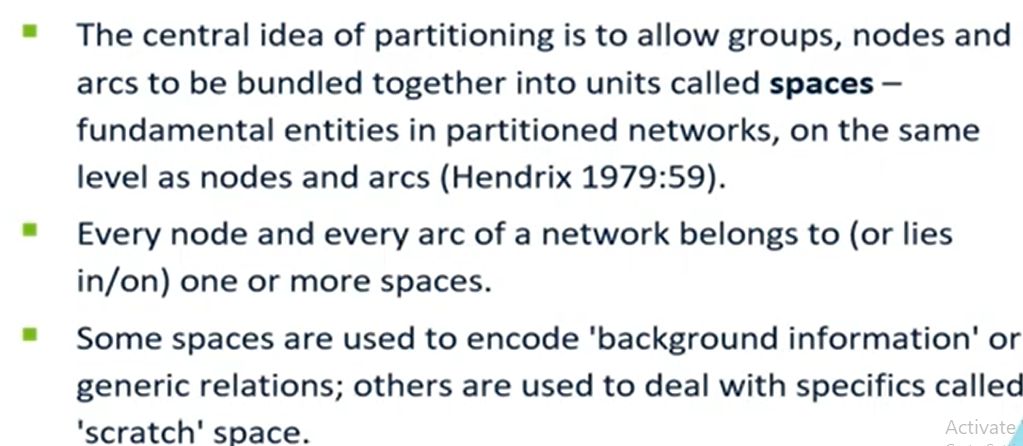
* Although inheritance is a useful way to express generalization about a class of objects, in some cases we need to express exceptions to those generalizations such as “Male do not give birth” or “Female dogs below the age of 6 months do not give birth”.

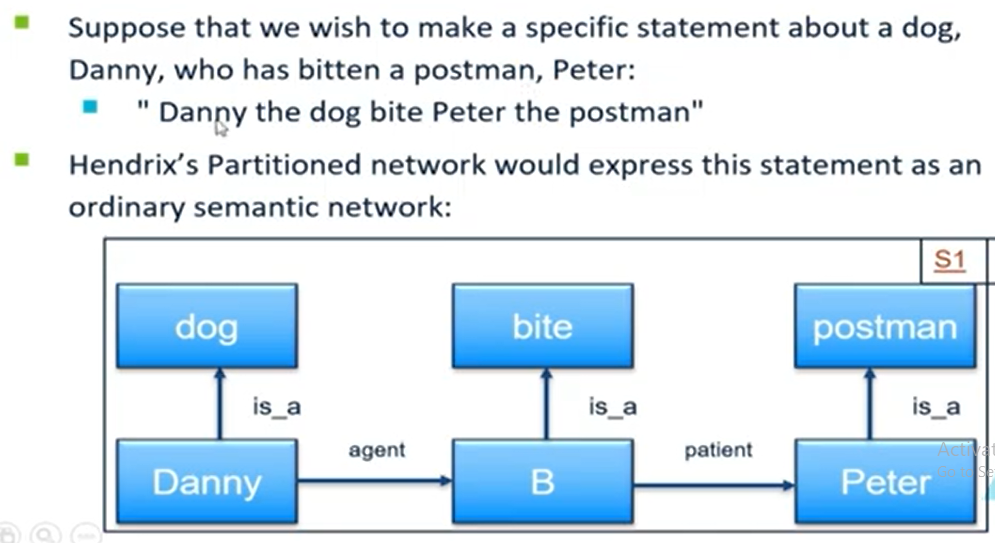


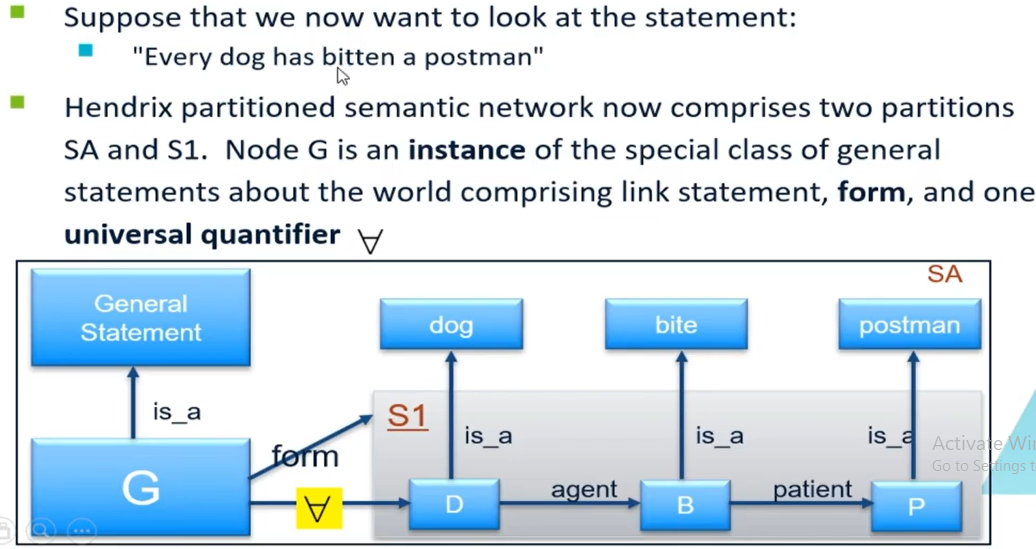
Extended Semantic Networks for KR:

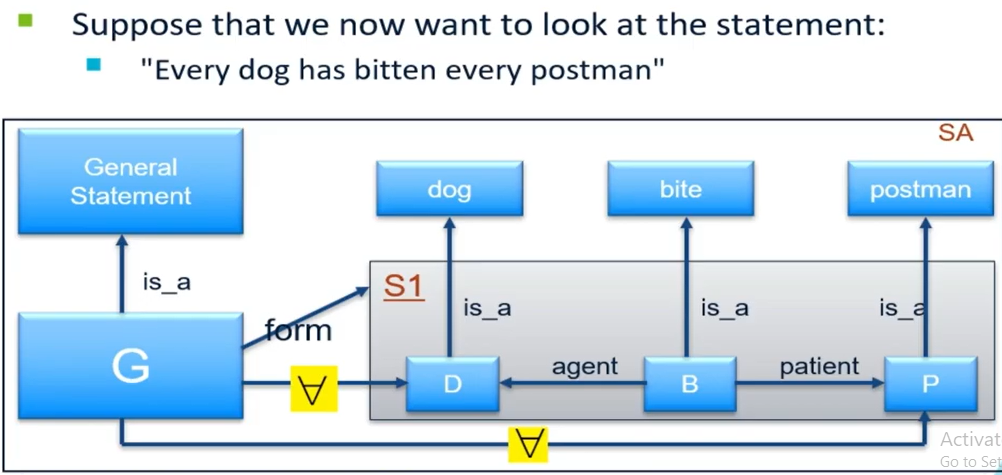


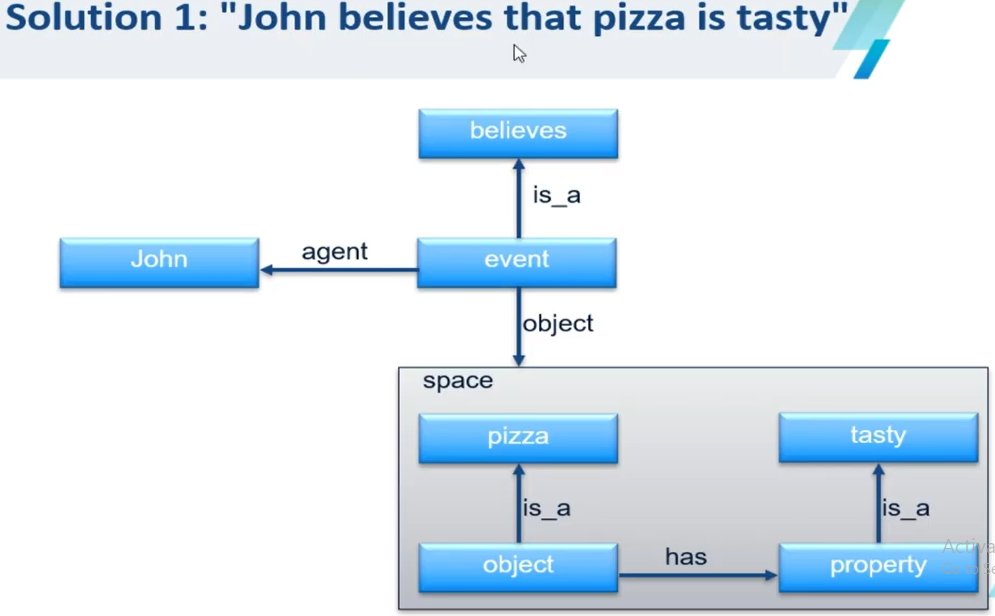


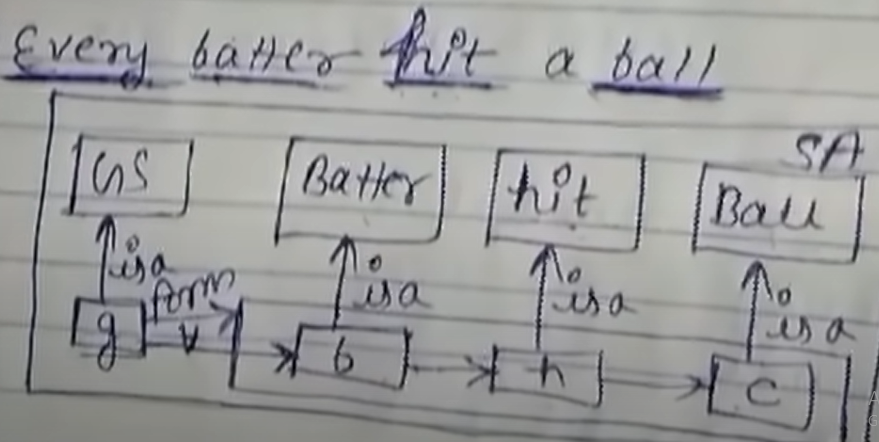












# C:\Users\user\Pictures\Screenshots\Screenshot (152).png

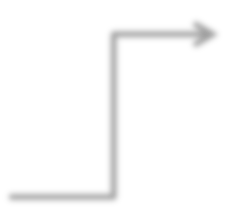
# Frames

Frame based representation is a development of semantic nets and allow us to express the idea of inheritance.

A Frame System consists of a set of frames (or nodes), which are connected together by relations. Each frame describes either an instance or a class. Each frame has one or more slots, which are assigned slot values. This is the way in which the frame system is built up. Rather than simply having links between frames, each relationship is expressed by a value being placed in a slot. Example:

|  |  |  |
| --- | --- | --- |
| **Frame Name** | **Slot** | **Slot Value** |
| Bob | is a | Builder |
| Owns | Tommy |
| Eats | Cheese |
| Tommy | is a | Dog |
| Chases | Bella |
| Bella | is a | Cat |
| Chases | Mice |
| Mice | Eat | Cheese |
|  |
| Builder |
| Dog |
| Cat |

We can also represent this frame system in a diagrammatic form such a below:



**Bob**

**Tommy**

is a Builder

is a

owns

Tommy

eats

chases

Bella

dog

cheese

When we say, “Tommy is a dog” we really mean, “Tommy is an instance of the class dog” or “Tommy is a member of the class dogs”.

## Why are Frames useful?

The main advantage of using frame-based systems for expert systems is that all information about a particular object is stored in one place.

## Inheritance in Frames:

Example: 1

We might extend our frame system with the following additional information: Dogs chase cats

Cats chase mice

Now, we do not need to state explicitly that Tommy chases Bella or that Tommy chases mice. We can inherit this information because Bella is an instance of the class cats, and Tommy is an instance of class dogs.

Example 2:

Mammals breathe Dogs are mammals Cats are mammals

Here we have created superclass mammals, of which dogs and cats are subclasses. Hence, we do not need to explicitly say that cats and dogs breathe because we can inherit this information. Similarly we do not need to express explicitly that Tommy and Bella breathe as they are instances of the class dogs and cats and therefore they inherit from those classes (superclasses).

Example 3: Let‟s take a fact:

Mammals have four legs

Now, this is not true as humans do not have four legs.

Or let‟s say Tommy has an unfortunate accident and now has only three legs. This information might be expressed as:

|  |  |  |
| --- | --- | --- |
| **Frame Name** | **Slot** | **Slot Value** |
| Mammals | \*no. of legs | Four |
| Dog | Subclass | Mammal |
| Cat | Subclass | Mammal |
| Tommy | is a | Dog |
| number of legs | Three |
| Bella | is a | Cat |

We have used an asterisk (\*) to indicate that the value for the “number of legs” slot for the mammal class is a default value and can be overridden.

## Slots as Frames:

* It is also possible to express a range of values that a slot can take. Ex, the no. of legs slot might be allowed a number between 1 & 4.
* Or we can express this restriction by allowing slots to be frames. The no. of legs slot can be represented as a frame such as.

|  |  |  |
| --- | --- | --- |
| **Frame Name** | **Slot** | **Slot Value** |
| No. of legs | Minimum value  Maximum value | 1  4 |

## Multiple Inheritance in Frame System:

It is possible for a frame to inherit properties from more than one frame, i.e. a class can be a subclass of two super classes and an object can be an instance of more than one class. This is known as **Multiple Inheritance**.

Example:

|  |  |  |
| --- | --- | --- |
| **Frame Name** | **Slot** | **Slot Value** |
| Human | Subclass | Mammal |
| No. of legs | Two |
| Builder | Builds | Houses |
| Bob | is a | Human |
| is a | Builder |

Hence, we can inherit the following information about Bob: He has two legs

He builds houses

In some cases, we encounter conflicts, where multiple inheritance leads us to conclude contradictory information about a frame.

Example:

|  |  |  |
| --- | --- | --- |
| **Frame Name** | **Slot** | **Slot Value** |
| Cheese | Is | Smelly |
| Things wrapped in foil | Is | not smelly |
| Cheddar | is a | Cheese |
| is a | Things wrapped in foil |

Here, cheddar is a type of cheese and that it comes wrapped in foil. Cheddar should inherit its smelliness from the cheese class, but it also inherits non- smelliness from the Things wrapped in foil class.

Thus, we need a mechanism to decide which features to inherit from which super class.

* + One simple method is to say that conflicts are resolved by the order in which they appear.

So, if a fact is established by instances and then that fact is contradicted by inheritance, the first fact is kept because it appeared first and the contradiction is discarded.

* + However, it would be better to build the frame system such that conflicts of this kind cannot occur.

### Conceptual Dependency (CD).

**In 1977, Roger C. Schank has developed a Conceptual Dependency structure. The Conceptual Dependency is used to represent knowledge of Artificial Intelligence. It should be powerful enough to represent these concepts of the sentence of natural language. It states that different sentence which has the same meaning should have some unique representation.**

**Disadvantage of semantic Nets: Representation of relation is in “is-a” form or “has”, which will conflict in representing the knowledge and called it as lack of formality (correct interpretation cannot be done)**

**Disadvantage of Frames: Modification of data in one frame can reflect the other frame and so has to modify the other frame**

**CD:**

• focuses on concepts instead of syntax.

• focuses on understanding instead of structure.

• introduced idea of a canonical meaning representation.

• different words and structures represent the same concept.

• language-independent meaning representation.

Canonical Meaning Representations

John gave Mary a book.

John gave a book to Mary.

Mary was given a book by John.

Mary took a book from John.

Mary received a book from John.

6 Primitive conceptual categories:

**PP**------Real world object (ex: I am speaking)

**ACT**----Real world action (ex: I am speaking)or(one of the 11 primitive actions)

**PA**------Attributes of objects (ex: I am an Engineer, youtuber, doctor,….)

**AA**------Attributes of actions(ex: actions done by an engineer, or doctor…..)

**T**---------Time (ex: I met her yesterday, meet me at 2:00PM)

**LOC**------Location (ex: I am in Pune)

Conceptual Roles

Conceptualization**:** The basic unit of the conceptual level of understanding.

• **Actor:** The performer of an ACT.

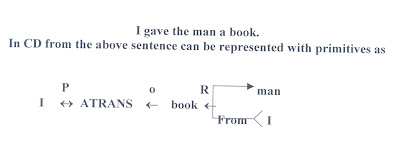
**• ACT:** An action done to an object.

• **Object:** A thing that is acted upon.

• **Recipient:** The receiver of an object as the result of an ACT.

• **Direction:** The location that an ACT is directed toward.

• **State:** The state that an object is in.

[](http://3.bp.blogspot.com/_ZGzaqHb40vU/TE6bvkys2zI/AAAAAAAAAIo/wgaYxrVNCQM/s1600/CD-Diagram.jpg)

**In the above representation the symbols have the following meaning:**

**Arrows ---------------**indicate direction of dependency

**Double arrow------** indicates two way link between actor and the action

**P-----------------------** indicates past tense

**ATRANS**-------------is one of the primitive acts used by the theory. It indicates transfer of

possession

**O----------------------** indicates the object case relation

**R----------------------** indicates the recipient case relation

Conceptual dependency provides a structure in which knowledge can be represented and also a set of building blocks from which representations can be built. A typical set of primitive actions are

ATRANS - Transfer of an abstract relationship(Eg: give,took)

PTRANS - Transfer of the physical location of an object(Eg: go)

PROPEL - Application of physical force to an object (Eg: push,pull,…..some forces)

MOVE - Movement of a body part by its owner (eg : kick ,movement of body parts by owner)

GRASP - Grasping of an object by an actor(Eg: throw)

INGEST - Ingesting of an object by an animal (Eg: eat,drink)

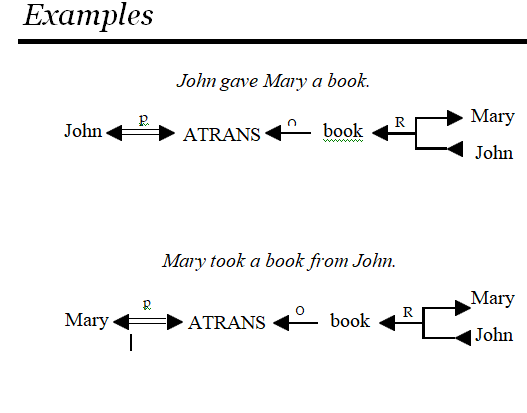
EXPEL - Expulsion of something from the body of an animal (cry,sweat,squeez)

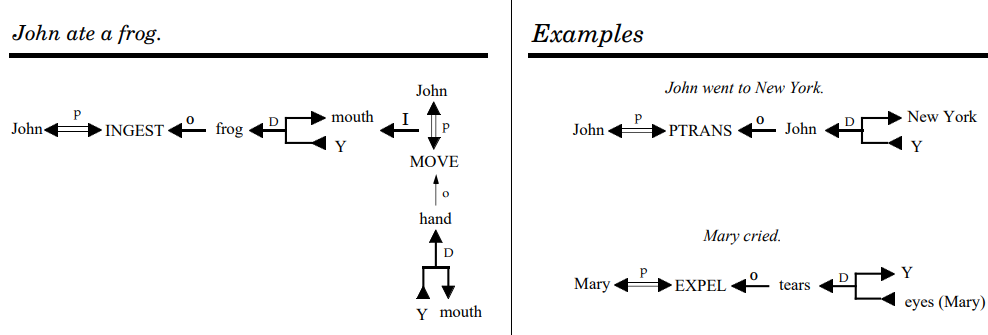
MTRANS - Transfer of mental information(Eg: tell,speak,see)

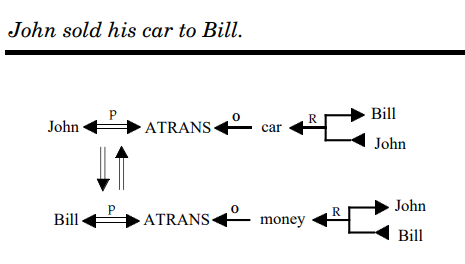
MBUILD - Building new information out of old(Eg: to decide, conclude)

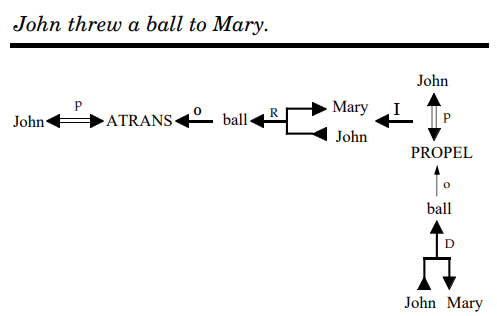
SPEAK - Production of sounds(Eg: play music, utter a sound)

ATTEND - Focusing of sense organ toward a stimulus (Eg: listen,see)









SCRIPTS:

A script is a structured representation describing a stereotyped(SOME ACTIVITIES) sequence of events in a particular context.

Scripts are used in **natural language understanding** systems to organize a knowledge base in terms of the situations that the system should understand. Scripts use a frame-like structure to represent the commonly occurring experience like going to the movies eating in a restaurant, shopping in a supermarket, or visiting an ophthalmologist.

Thus, a script is a structure that prescribes a set of circumstances that could be expected to follow on from one another.

**Scripts are beneficial because:**

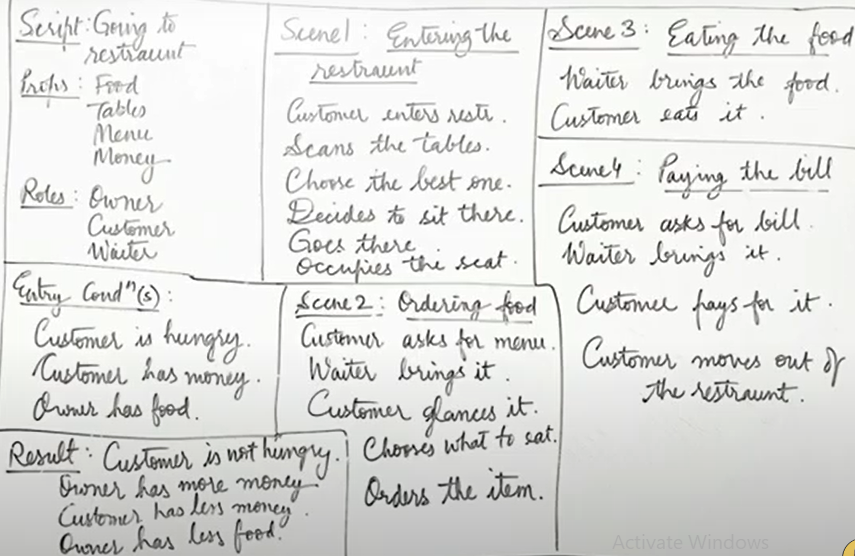
* Events tend to occur in patterns.
* A casual relationship between events exist.
* An entry condition exists which allows an event to take place.
* Prerequisites exist upon events taking place.

Components of a script

The components of a script include:

* **Entry condition:** These are basic condition which must be fulfilled before events in the script can occur.
* **Results:**Condition that will be true after events in script occurred.
* **Props:**Slots representing objects involved in events
* **Roles:**These are the actions that the individual participants perform.
* **Track:**Variations on the script. Different tracks may share components of the same scripts.
* **Scenes:**The sequence of events that occur.

PSEUDO SCRIPT FOR GOING TO RESTAURANT:

****

Describing a script, special symbols of actions are used. These are:

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Meaning** | **Example** |
| ATRANS | transfer a relationship | Give |
| PTRANS | transfer physical location of an object | Go |
| PROPEL | apply physical force to an object | Push |
| MOVE | move body part by owner | Kick |
| GRASP | grab an object by an actor | Hold |
| INGEST | taking an object by an animal eat | Drink |
| EXPEL | expel from animal’s body | Cry |
| MTRANS | transfer mental information | Tell |
| MBUILD | mentally make new information | Decide |
| CONC | conceptualize or think about an idea | Think |
| SPEAK | produce sound | Say |
| ATTEND | focus sense organ | Listen |

**Example:-Script for going to the bank to withdraw money.**

**SCRIPT** : Withdraw money

**TRACK** : Bank

**PROPS** : Money

Counter

Form

Token

**Roles** : P= Customer

E= Employee

C= Cashier

**Entry conditions**: P has no or less money.

The bank is open.

**Results** : P has more money.

**Scene 1:**Entering

P PTRANS P into the Bank

P ATTEND eyes to E

P MOVE P to E

**Scene 2:**Filling form

P MTRANS signal to E

E ATRANS form to P

P PROPEL form for writing

P ATRANS form TO E

**Scene 3:**Withdrawing money

P ATTEND eyes to counter

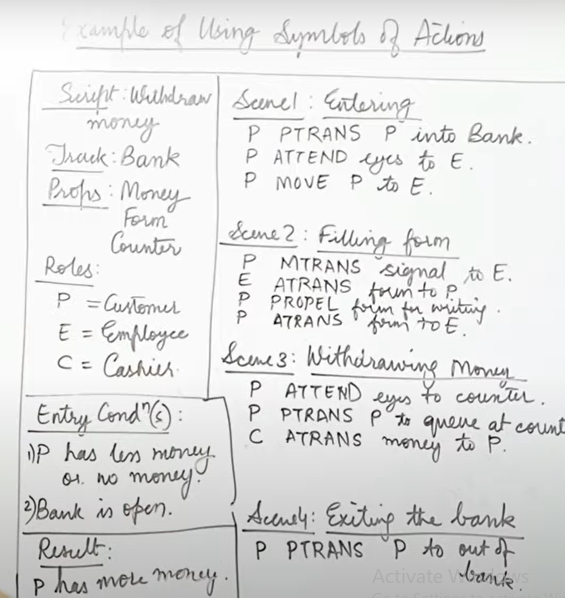
P MOVE P to queue at the counter

P ATRANS token to C

C ATRANS money to P

**Scene 4:**Exiting the bank

P PTRANS P to out of bank



**Advantages of Scripts**

* Ability to predict events.
* A single coherent interpretation maybe builds up from a collection of observations.

**Disadvantages of Scripts**

* Less general than frames.
* May not be suitable to represent all kinds of knowledge