UNIT III KNOWLEDGE INFERENCE
Knowledge representation -Production based system, Frame based system. Inference - Backward chaining, Forward chaining, Rule value approach, Fuzzy reasoning - Certainty factors, Bayesian Theory-Bayesian Network-Dempster - Shafer theory.

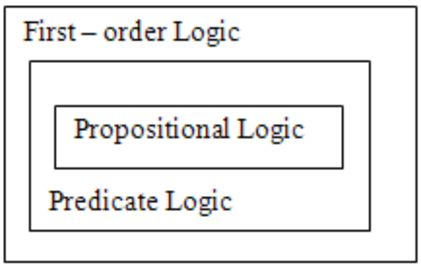
Types of Knowledge Representation

Knowledge can be represented in different ways. The structuring of knowledge and how designers might view it, as well as the type of structures used internally are considered. Different knowledge representation techniques are:

a. Logic
b. Semantic Network
c. Frame
d. Conceptual Graphs
e. Conceptual Dependency
f. Script

a.Logic

A logic is a formal language, with precisely defined syntax and semantics, which supports sound inference. Different logics exist, which allow you to represent different kinds of things, and which allow more or less efficient inference. The logic may be different types like propositional logic, predicate logic, temporal logic, description logic etc. But representing something in logic may not be very natural and inferences may not be efficient.



Weak Slot-And-Filler structures includes

- **Semantic net**
- **Frames**

b. Semantic Network

A semantic network is a graphical knowledge representation technique. This knowledge representation system is primarily on network structure. The semantic networks were basically developed to model human memory.

In a **semantic net**, information is represented as a set of nodes connected to each other by a set of labeled **arcs**, which represent relationship among the nodes. Each **node** should contain the information about **objects** and each **arc** should contain the **relationship between objects**.

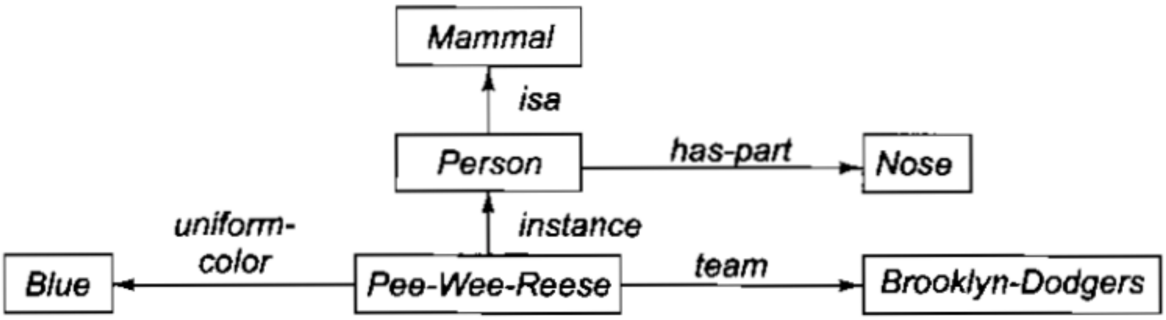


Fig *A Semantic Network*

This network contains examples of both **isa** and **instance** relations, as well as some other, more domain specific relations like **team** and **uniform color.**

Using inheritance additional relation can also be derived: has-part(Pee-Wee-Reese, Nose).

b.1 Intersection Search

Semantic nets are used to find relationships among objects by spreading activation about from each of two nodes and seeing where the activation met this process is called **intersection search.**

Using this answer for questions like – what is the relation between Brooklyn-Dodgers and Blue.

Advantage of Slot-and-Filler structure: provides representation entity based organization of knowledge.

b.2 Representing Non-binary Predicates

Simple **binary predicates** like:

- *isa*(Person,Mammal)
- *instance*(Pee-Wee-Reese, person)
- *team*(Pee-Wee-Reese, Brooklyn-Dodgers)
- *uniform color*(Pee-Wee-Reese, Blue)

can be represented easily by semantic nets. And other **unary predicates** can also be represented by using general-purpose predicates such as isa and instance.

- **man(Marcus)** can be rewritten as **instance(Marcus,Man)**

Three or even more place predicates can also be converted to a binary form by creating one new object representing the entire predicate statement and then introducing binary predicates to describe relationship to this new object.

Example: score(Cubs, Dodgers, 5-3)

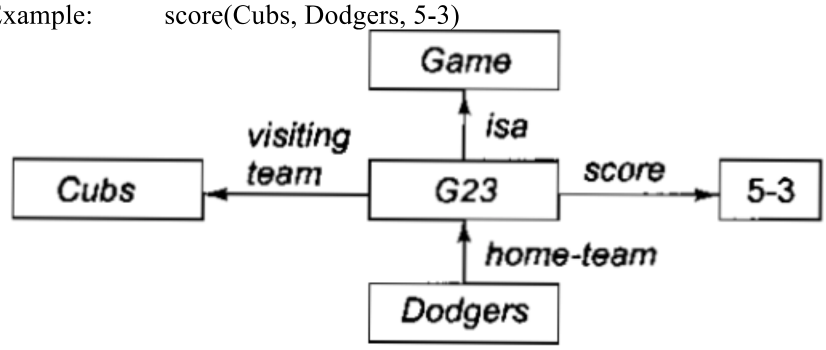


Fig. *A Semantic Net for an n-Place Predicate*

This can be represented in a semantic net by creating a node to represent the specific game and then relating each of thr three pieces of information to it.

This technique is particularly useful for representing the contents of a typical declarative sentence that describes several aspects of a particular event.

Eg: John gave the book to Mary

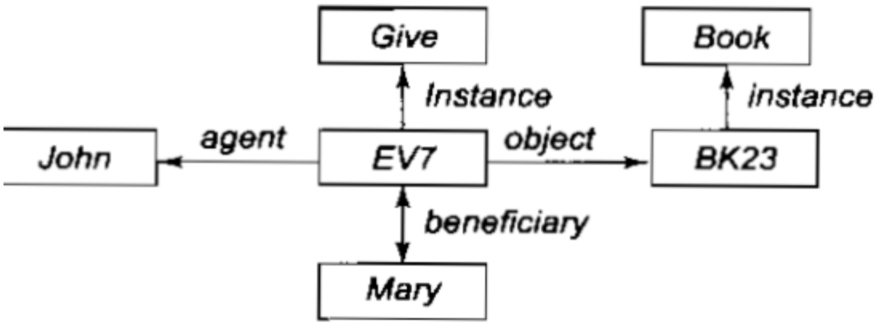
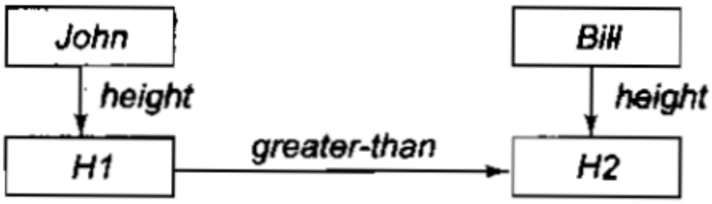


Fig. *A Semantic Net Representing a Sentence*

Uses: English understanding programs.

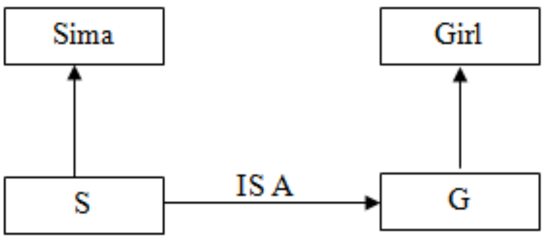
b.3 Making important distinctions

These entities are independent of each other, now we want to prove that John is taller than Bill – then we do this :



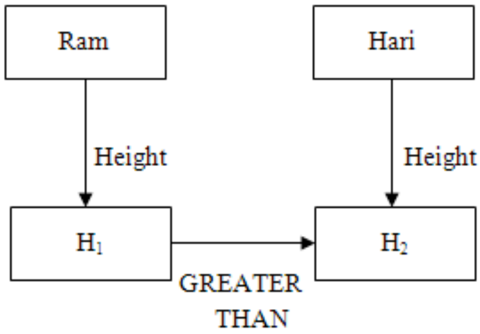
Some more examples:

1. Suppose we have to represent the sentence "Sima is a girl".

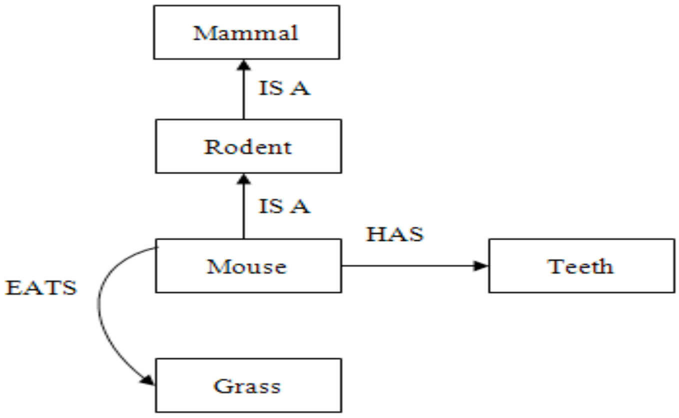


2. Ram is taller than Hari

   It can also be represented as



3. “Mouse is a Rodent and Rodent is a mammal. Mouse has teeth and etas grass”. Check whether the sentence mammal has teeth is valid or not.]



b.4 Partitioned Semantic Network

Some complex sentences are there which cannot be represented by simple semantic nets and for this we have to follow the technique partitioned semantic networks. Partitioned semantic net allow for

1. Propositions to be made without commitment to truth.
2. Expressions to be quantified.

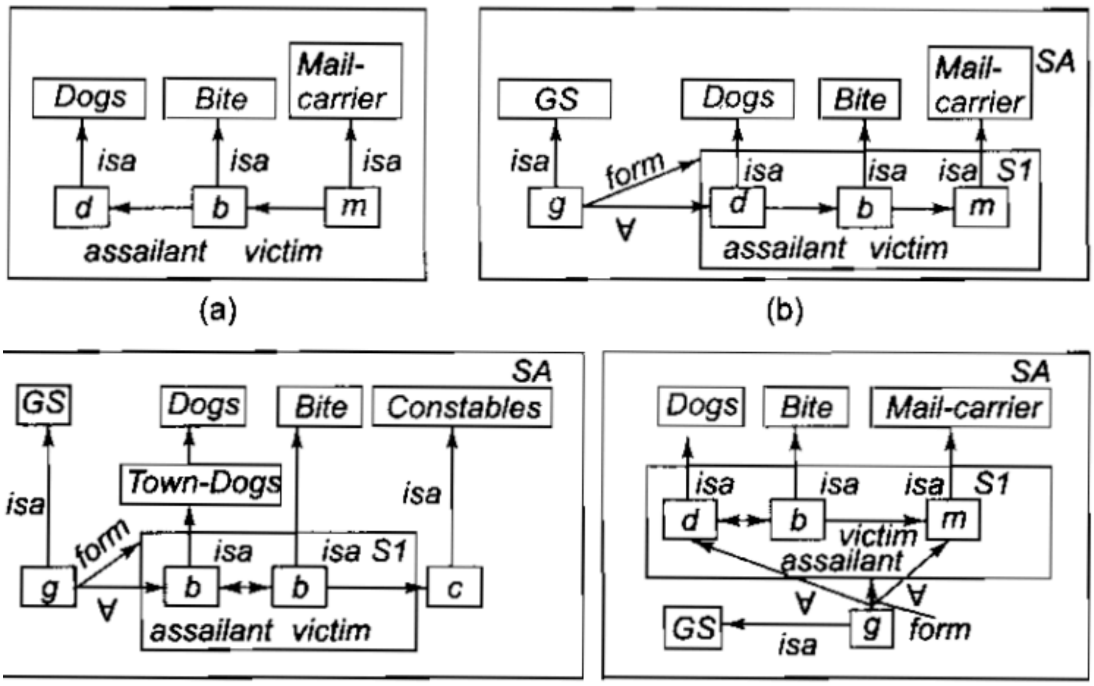
In partitioned semantic network, the network is broken into spaces which consist of groups of nodes and arcs and regard each space as a node.

Fig a. Corresponds to the statement **“ The dog bit the mail carrier” .** Here the nodes Dogs, bite, mail-carrier represent the classes of dogs, bitting and mail-carrier respectively, while the nodes d, b and m represent a particular dog, a particular bitting and a particular mail-carrier. This fact can easily be represented by a single net with no partitioning.

Whereas the fact **“ Every dog has bitten a mail carrier” or in logic as**

$$\forall x : Dog(x) \rightarrow \exists y : Mail\text{-}Carrier\ (y) \wedge Bite \wedge (x, y)$$

should be done using partitioning only as shown in **Fig b.** Where node g is an instance of the special class GS of general statement about the world(ie those with universal quantifiers).

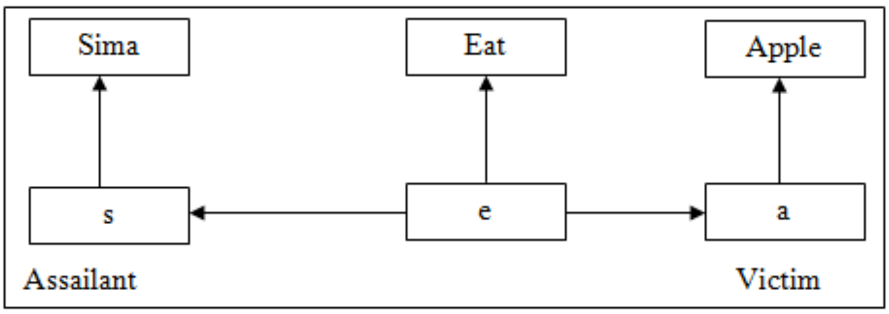


(a) (b) (c) (d)

Let us consider few examples.

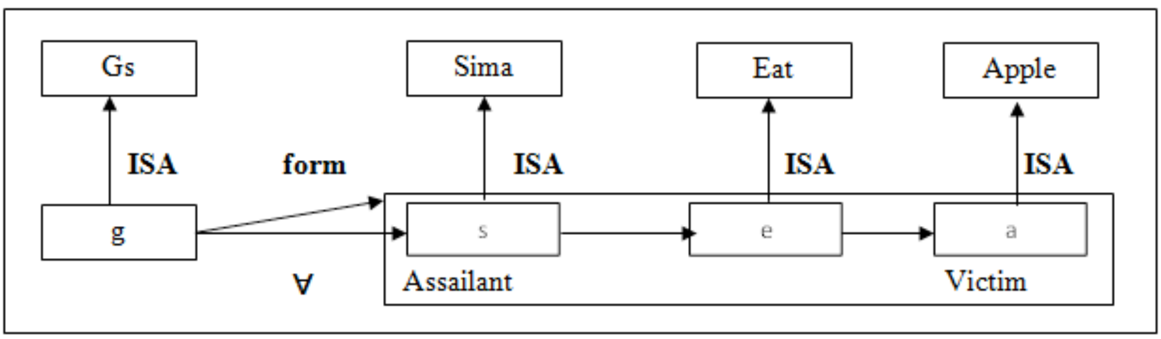
Draw the partitioned semantic network structure for the followings:

a) Sima is eating an apple.

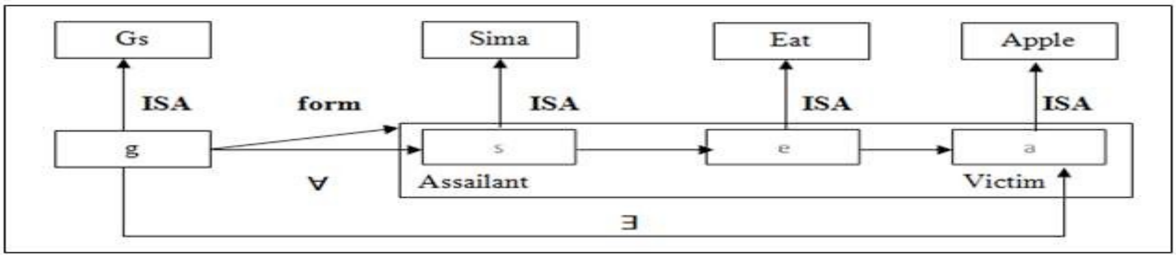


Figure

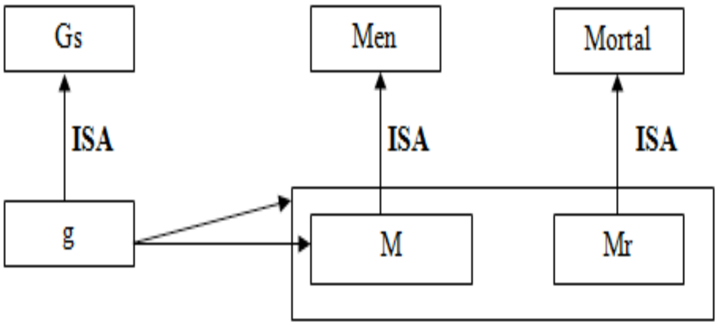
b) All Sima are eating an apple.



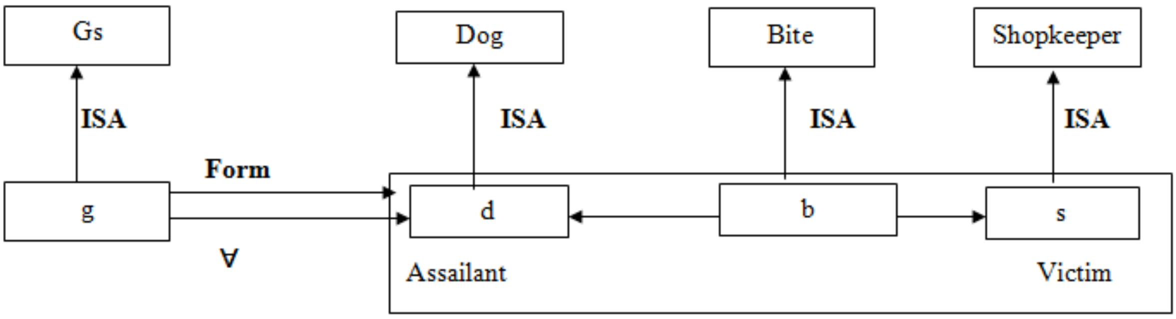
b) All Sima are eating some apple.



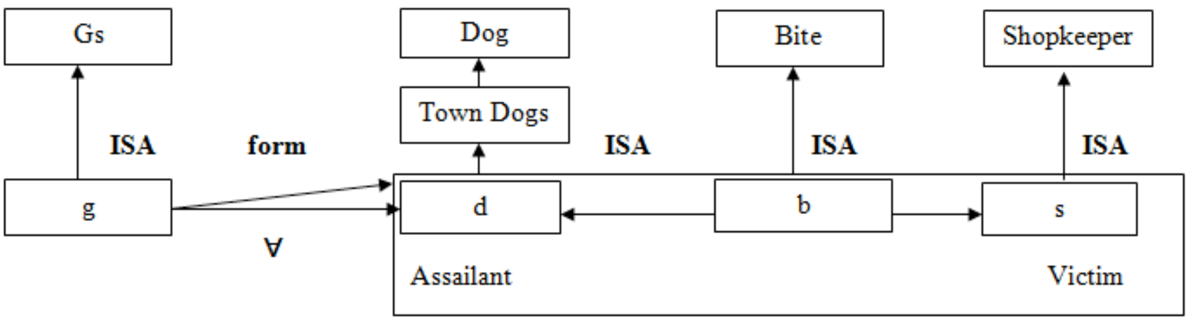
c) All men are mortal



d) Every dog has bitten a shopkeeper



e) Every dog in town has bitten a shopkeeper.



NOTE: On the above semantic network structures, the instance "IS A" is used. Also two terms like assailant and victim are used. Assailant means "by which the work is done" and that of victim refers to that states which a relation is being asserted. Secondly, one or more for all (∀) or there exists (∃) connections which represent universally quantifiable variables.

Evolution of Frames

As seen in the previous problem, there are problems which are difficult to solve with Semantic Nets. Although there is no clear distinction between a semantic net and frame system, then more structure the system has, the more likely it is to be termed a frame system.

c. Frames

A frame is a collection of attributes and associated values that describe some entity in the world.

- Frames are general record like structures which consist of a collection of slots and slot values.
- The slots may be of any size and type.
- Slots typically have names and values or subfields called facets. Facets may also have names and any number of values.
- A frame may have any number of slots, a slot may have any number of facets, each with any number of values.
- A slot contains information such as attribute value pairs, default values, condition for filling a slot, pointers to other related frames and procedures that are activated when needed for different purposes.
- Sometimes a frame describes an entity in some absolute sense, sometimes it represents the entity from a particular point of view.
- A single frame taken alone is rarely useful.
- We build frame systems out of collection of frames that are connected to each other by virtue of the fact that the value of an attribute of one frame may be another frame.

Syntax of a frame

```
(<frame name>
(<slot1> (<facet1> <value 1>..........<value n1>)
         (<facet2> <value1>............<value n2>)
         .
         .
         .
         .
         .
         (<facet n> <value1>............ <value nn>))
(<slot 2> (<facet1> <value 1>.....<value n1>)
         (<facet2><value2>...........<value n2>)
         .
         .
         ))
```

c.1 Frames as Sets and Instances

The set theory is a good basis for understanding frame systems.

Each frame represents either a class (a set) or an instance (an element of class)

Considering the Cricket example – Person, Adult Male, Bowler, Team are all classes. The frames Pee-Wee-Pee-Wee-Reese and Brooklyn-Dodgers are instances.

isa relation used to represent subset of the set.

Eg: the set of adult-male is a subset of the set of people

Instance relation corresponds to the relation element of.

Eg: Pee-wee-Reese is an element of the set of fielders

Both isa and instance relations have inverse attributes, which we call subclasses & all-instances.

As a class is represents a set, there are 2 kinds of attributes that can be associated with it.

Its own attributes & Attributes that are to be inherited by each element of the set.(* - This is how they are represented.)

Fig below shows a simplified Frame system

```
Person
    isa :                   Mammal
    cardinality :           6,000,000,000
    * handed :              Right
Adult-Male
    isa :                   Person
    cardinality :           2,000,000,000
    * height :              5-10
ML-Baseball-Player
    isa :                   Adult-Male
    cardinality :           624
    *height :               6-1
    * bats :                equal to handed
    * batting-average :     .252
    * team :
    * uniform-color :
Fielder
    isa :                   ML-Baseball-Player
    cardinality :           376
    *batting-average :      .262
Pee-Wee-Reese
    instance :              Fielder
    height :                5-10
    bats :                  Right
    batting-average :       .309
    team :                  Brooklyn-Dodgers
    uniform-color :         Blue
ML-Baseball-Team
    isa:                    Team
    cardinality :           26
    * team-size :           24
    * manager :
Brooklyn-Dodgers
    instance :              ML-Baseball-Team
    team-size :             24
    manager :               Leo-Durocher
```

Representing meta classes as frames

- Meta classes are special classes whose elements are themselves classes.
- A class is now an element of (instance) some class as well as a subclass (isa) of one or more classes
- A class inherits properties from the class of which it is an instance, just as any instance does.
- In addition, a class passes inheritable properties down from its superclasses to its instances .

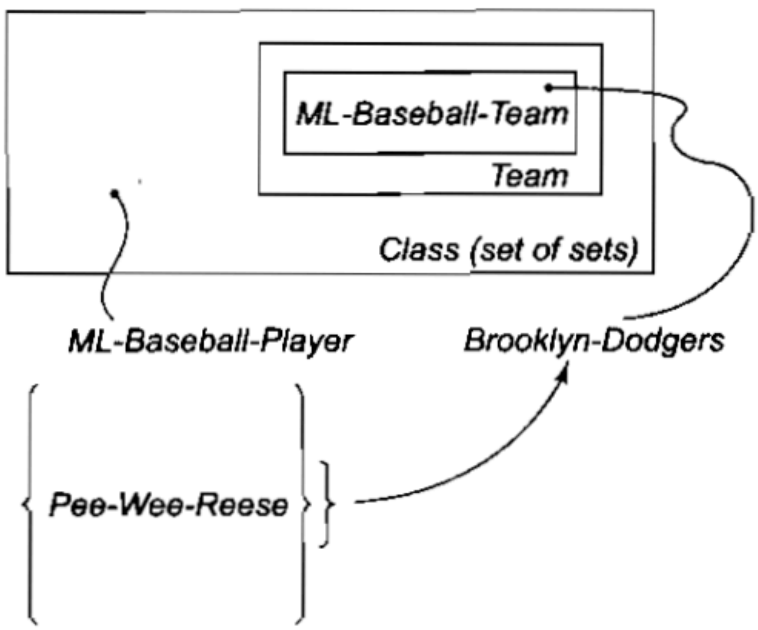
Class
- instance : Class
- isa : Class
- * cardinality :

Team
- instance : Class
- isa : Class
- cardinality : {the number of teams that exist}
- *team-size : {each team has a size}

ML-Baseball-Team
- isa : Mammal
- instance : Class
- isa : Team
- cardi nality : 26 {the number of baseball teams that
- * team-size : 24 {default 24 players on a team}
- * manager :

In the fig. given below classes and meta classes.

- Class is the most basic meta class.
- All other classes are instances of it, either directly or through one of its subclasses
- In the Fig. **Team** is a **subclass(subset)** of Class and **ML-Baseball-team is a subclass of Team**
- The class CLASS introduces the attribute cardinality, which is to be inherited by all instances of CLASS.
- Cardinality stands for number.

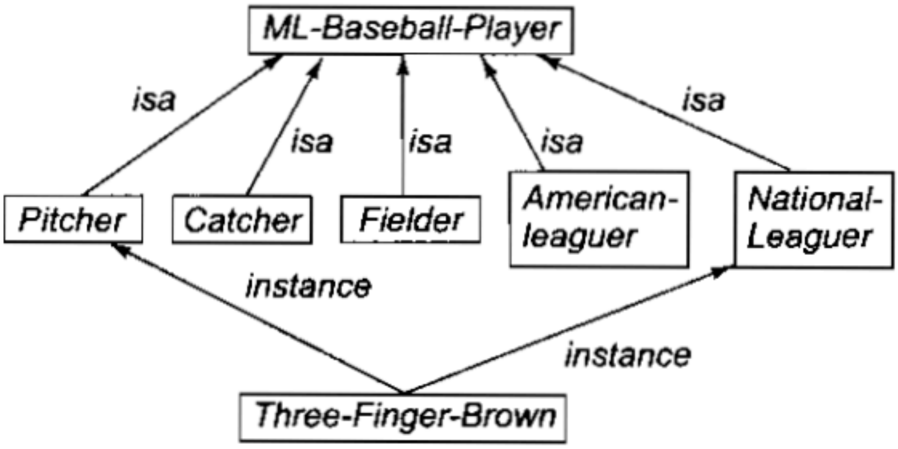


Other ways of Relating Classes to Each Other

- A class1 can be a subset of class2.

are guaranteed to have no elements in common with it.

- Another one is – is-covered-by :- which relates a class to a set of subclasses, the union of which is equal to it.
- If a class is-covered-by a set S of mutually disjoint classes, then S is called a partition of the class.




```
ML-Baseball-Player
    is-covered-by :            {Pitcher, Catcher, Fielder}
                               {American-Leaguer, National-Leaguer
Pitcher
    isa :                      ML-Baseball-Player
    mutually-disjoint-with :   {Catcher, Fielder}
Catcher
    isa :                      ML-Baseball-Player
    mutually-disjoint-with:    {Pitcher, Fielder}
Fielder
    isa :                      ML-Baseball-Player
    mutually-disjoint-with :   {Pitcher, Catcher}
American-Leaguer
    isa :                      ML-Baseball-Player
    mutually-disjoint-with :   {National-Leaguer}
National-Leaguer
    isa :                      ML-Baseball-Player
    mutually-disjoint-with :   {American-Leaguer}
Three-Finger-Brown
    instance :                 Pitcher
    instance :                 National-Leaguer
```

Slots as Full-Fledged Objects(Frames)

- Some of the properties able to represent and use in reasoning include :
    - The class to which the attribute can be attached? Constraints on either the type or the value of the attribute.
    - A default value for the attribute.
    - Rules for inheriting values for the attribute.

- These frames will be organized into an isa hierarchy, just as any other frames are, and that hierarchy can then be used to support inheritance of values for attributes of slots.

Now let us formalize what is a slot.

- A slot here is talked about as a relation . It maps from elements of its domain (the classes for which it makes sense) to elements of its range(its possible values). A relation is a set of ordered pairs. Thus it makes sense to say that relation R1 is a subset of another R2.
- In that case R1 is a specialization of R2.
- Since a slot is a set , the set of all slots, which we will call SLOT, is a meta class. Its instances are slots, which may have sub slots

Slots Values as Objects

Using slot "values" as objects. Let's take the following example –

- John :
  - height : 72
- Bill :
  - height :
- The only information we have here is that John is taller than Bill

Let us consider the below examples.

1) Create a frame of the person Ram who is a doctor. He is of 40. His wife name is Sita. They have two children Babu and Gita. They live in 100 kps street in the city of Delhi in India. The zip code is 756005.

```
Ram
   isa:            Doctor
   age :           40
   wife:           Sita
   children1:      Babu
   children2:      Gita
   address1:       100 kps street
   city:           Delhi
   country:        india
   zip:            756005
```

Eg. FOR OTHER WAY OF WRITING FRAMES

Create a frame of the person Anand who is a chemistry professor in RD Women's College. His wife name is Sangita having two children Rupa and Shipa.

```
(Anand
        (PROFESSION (VALUE Chemistry Professor))
        (ADDRESS (VALUE RD Women's College))
        (WIFE (VALUE Sangita)) (CHILDREN(VALUE
        RupaShipa)))
```

Create a frame of the person Akash who has a white maruti car of LX-400 Model. It has 5 doors. Its weight is 225kg, capacity is 8, and mileage is 15 km /lit.

```
(Akash
   (CAR (VALUE Maruti))
   (COLOUR (VALUE White))
   (MODEL (VALUE LX-400))
   (DOOR (VALUE 5))
```

(CAPACITY (VALUE 8))
(MILAGE (VALUE 15km/lit)))

The frames can be attached with another frame and can create a network of frames. The main task of action frame is to provide the facility for procedural attachment and help in reasoning process.

Reasoning using frames is done by instantiation. Instantiation process begins, when the given situation is matched with frames that are already in existence. The reasoning process tries to match the current problem state with the frame slot and assigns them values. The values assigned to the slots depict a particular situation and by this, the reasoning process moves towards a goal. The reasoning process can be defined as filling slot values in frames.

d. Conceptual Graphs

It is a knowledge representation technique which consists of basic concepts and the relationship between them.

- As the name indicates, it tries to capture the concepts about the events and represents them in the form of a graph.
- A concept may be individual or generic. An individual concept has a type field followed by a reference field.
- For **example person : Ram**. Here person indicates type and Ram indicates reference.
- An individual concept should be represented within a rectangle in graphical representation and within a square bracket in linear representation.
- The generic concept should be represented within an oval in graphical representation and within a parenthesis in linear representation.
- Conceptual graph is a basic building block for associative network.
- Concepts like AGENT, OBJECT, INSTRUMENT, PART are obtained from a collection of standard concepts. New concepts and relations can be defined from these basic ones.
- These are also basic building block for associative network.
- A linear conceptual graph is an elementary form of this structure.
- A single conceptual graph is roughly equivalent to a graphical diagram of a natural language sentence where the words are depicted as concepts and relationships.

Consider an example

"Ram is eating an apple "

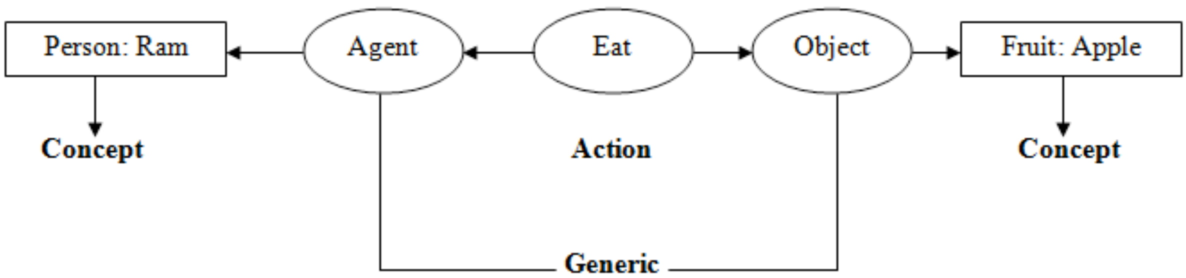


Figure Graphical Representation

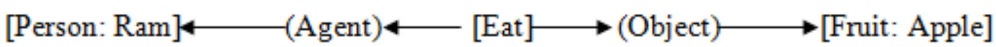


Figure Linear Representation

Strong Slot-And-Filler structures includes

- Conceptual Dependency

- **Scripts**

e. Conceptual Dependency

- It is an another knowledge representation technique in which we can represent any kind of knowledge.
- Conceptual dependency theory is based on the use of knowledge representation methodology was primarily developed to understand and represent natural language structures.
- The conceptual dependency representation captures maximum concepts to provide canonical form of meaning of sentences.

Generally there are four primitives from which the conceptual dependency structure can be described. They are

a. ACTS : Actions

b. PPs : Objects (Picture Producers)

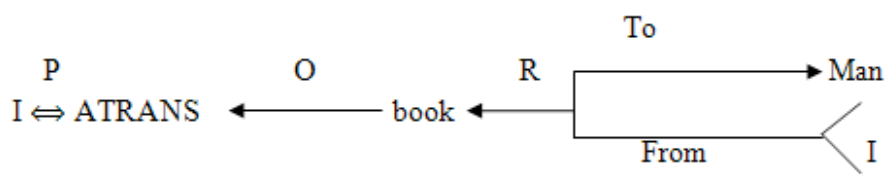
c. AAs : Modifiers of Actions (Action Aiders)

d. PAs : Modifiers of PPs (Picture Aiders)

e. TS : Time of action

Conceptual dependency provides both a structure and a specific set of primitives at a particular level of granularity, out of which representation of particular pieces of information can be constructed.

For example



Where ←: Direction of dependency

Double arrow indicates two way link between actor and action.
P: Past Tense
ATRANS: One of the primitive acts used by the theory
O: The objective case relation
R: Recipient case Relation

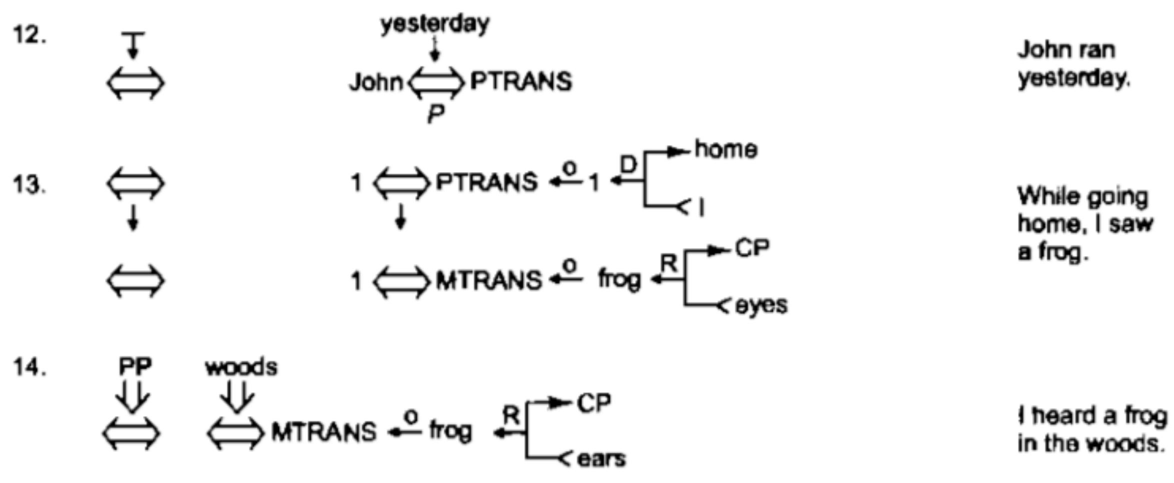
In CD, representation of actions are built from a **set of primitive acts**.

1) **ATRANS:** Transfer of an abstract relationship (give, accept, take)
2) **PTRANS:** Transfer the physical location of an object (Go, Come, Run, Walk)
3) **MTRANS:** Transfer the mental information (Tell)
4) **PROPEL:** Application of physical force to an object (push, pull, throw)
5) **MOVE:** Movement of a body part by its owner (kick).
6) **GRASP:** Grasping of an object by an action (clutch)
7) **INGEST:** Ingestion of an object by an animal (eat)
8) **EXPEL:** Expel from an animal body (cry)
9) **MBUILD:** Building new information out of old (decide)
10) **SPEAK:** Production of sounds (say)

The main goal of CD representation is to capture the implicit concept of a sentence and make it explicit. In normal representation of the concepts, besides actor and object, other concepts of time, location, source and destination are also mentioned. Following conceptual tenses are used in CD representation.

1) O : Object case relationship
2) R : Recipient case relationship
3) P : Past
4) F : Future
5) Nil : Present
6) T : Transition
7) Ts : Start Transition
8) Tf : Finisher Transition
9) K : Continuing
10) ? : Interrogative
11) / : Negative
12) C : Conditional

Also there are several rules in conceptual dependency



Rule 1: PP $\Longleftrightarrow$ ACT

It describes the relationship between an actor and an event, he/she causes.
E.g. Ram ran

Ram $\overset{P}{\Longleftrightarrow}$ PTRANS
Where P: Past Tense

Rule 2: PP $\Longleftrightarrow$ PA

It describes the relationship between a PP and PA where the PA indicates one characteristics of PP. E.g. Ram is tall

Ram $\overset{Nil}{\Longleftrightarrow}$ Tallor Ram $\overset{Nil}{\Longleftrightarrow}$ He ight (> Average)

Rule 3: PP $\Longleftrightarrow$ PP

It describes the relationship between two PPs where one PP is defined by other.
E.g. Ram is a doctor

Ram $\overset{Nil}{\Longleftrightarrow}$ Doctor

Rule 4: PP or PA

$$\begin{array}{ccc} PP & \text{or} & PA \\ \uparrow & & \downarrow \\ PA & & PP \end{array}$$

It describes the relationship between the PP and PA, where PA indicates one attributes of PP.
E.g. A nice boy is a doctor

$$\begin{array}{cc} Boy & \overset{Nil}{\Longleftrightarrow} Doctor \\ \uparrow & \\ Nice & \end{array}$$

Rule 5: PP

$\Uparrow$

PP

It describes the relationship between 3 PP's where one PP is the owner of another PP.

E.g. Ram's Cat

Cat

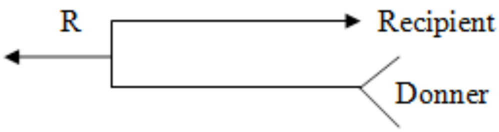
$\Uparrow$

Ram

Rule 6: Act $\xleftarrow{O}$ PP Where O: Object

It describes the relationship between the PP and ACT. Where PP indicates the object of that action. E.g. Ram is eating an apple.

Ram $\Longleftrightarrow$ INGEST $\xleftarrow{O}$ Apple

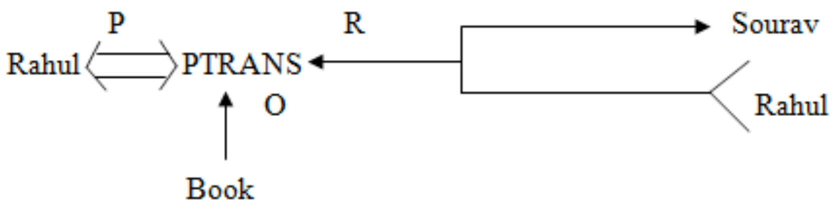


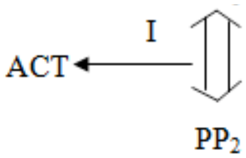
Rule 7: ACT

(R: Recipient)

Here one PP describes the recipient and another PP describes the donner

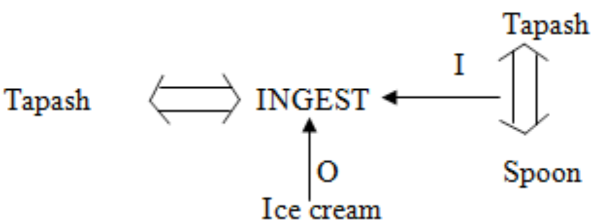
E.g. Rahul gave a book to sourav.



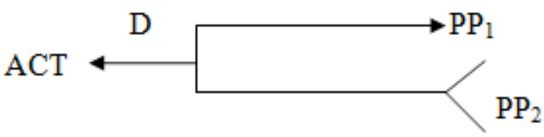
Rule 8:  (I: Instrument used in the action)

Here PP_1 indicates the agent and PP_2 indicates the object that is used in the action.

E.g. Tapash ate the ice cream with the spoon.

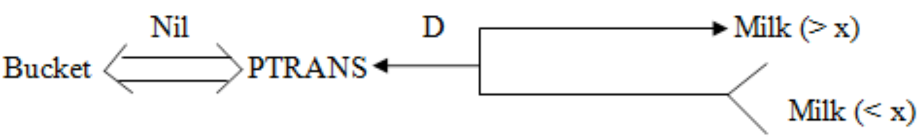


Rule 9:



Here D indicates destination, PP_1 indicates destination and PP_2 indicates the source.

E.g. the bucket is filled with milk.



xindicates the average milk and the source i.e. bucket is dry which is hidden

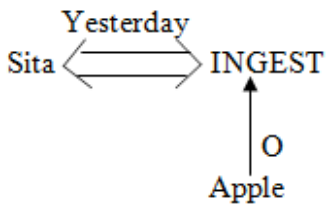
Rule 10:

$$PP \overset{T}{\Longleftrightarrow} ACT$$

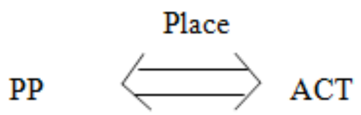
(T: Time)

It describes the relationship between a conceptualization and the time at which the event is described occurs.

E.g. Sita ate the apple yesterday.

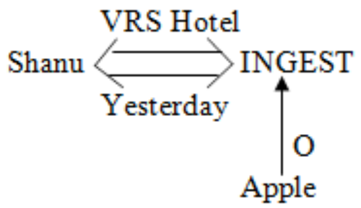


Rule 11:

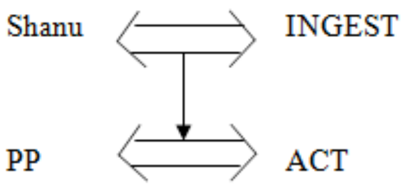


It describes the relationship between a conceptualization and the place at which it is   occurred.

E.g. Shanu ate the apple at VRS hotel yesterday

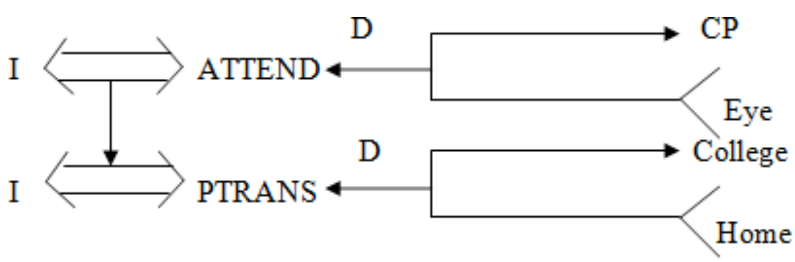


Rule 12:



It describes the relationship between one conceptualization with another.

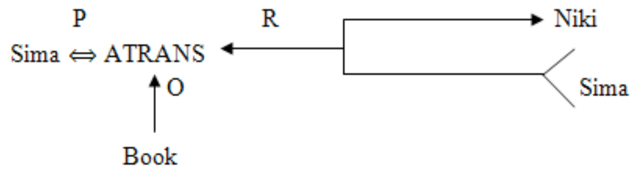
E.g. while I was going to college, I saw a snake



(Where CP: Conscious Processor i.e. the combination of all sense organs like eye, ear, nose   etc.)

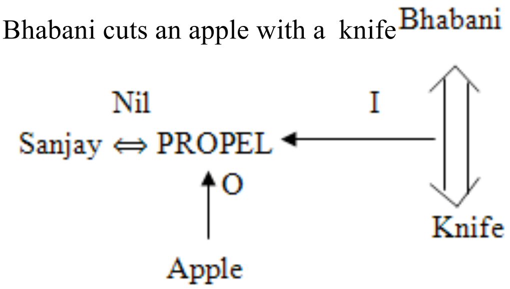
By using the above rules we can represent any sentence. Let us visualize few examples on conceptual dependency.

1) Sima gave a book to Niki

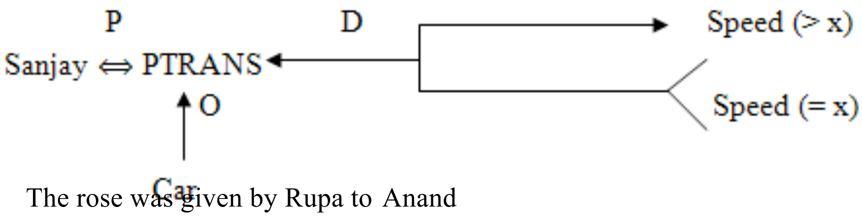


Where O: Object, P: Past Tense, R: Recipient, Sima: PP, Book: PP, Niki: PP, ATRANS: give

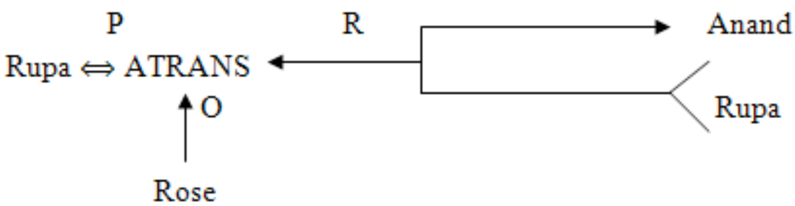
2) Bhabani cuts an apple with a knife



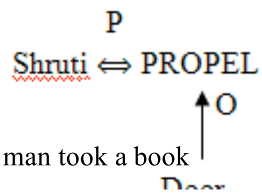
3) Sanjay drove the car fast



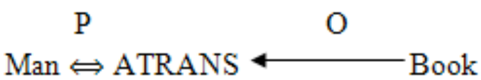
4) The rose was given by Rupa to Anand



5) Shruti pushed the door.



6) The man took a book

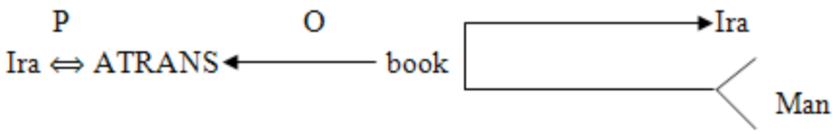


Here man is the doctor and book is the object of the action took.

7) My grandfather told me a story

$$\text{Grandfather} \overset{P}{\Leftrightarrow} \text{MTRANS} \xleftarrow{\quad O \quad} \text{Story}$$

8) Ira gave the man a dictionary



f. <u>SCRIPT</u>

- It is an another knowledge representation technique.
- Scripts are frame like structures used to represent commonly occurring experiences such as going to restaurant, visiting a doctor.
- A script is a structure that describes a stereotyped sequence of events in a particular context.
- A script consist of set of slots. Associated with each slot may be some information about what kinds of values it may contain as well as a default value to be used if no other information is available.
- Scripts are useful because in the real world, there are no patterns to the occurrence of events.
- These patterns arise because of clausal relationships between events.
- The events described in a script form a giant casual chain.
- The beginning of the chain is the set of entry conditions which enable the first events of the script to occur.
- The end of the chain is the set of results which may enable later events to occur. The headers of a script can all serve as indicators that the script should be activated.
- Once a script has been activated, there are a variety of ways in which it can be useful in interpreting a particular situation. A script has the ability to predict events that has not explicitly been observed.

An important use of scripts is

- To provide a way of building a single coherent interpretation from a collection of observation.
- Scripts are less general structures than are frames and so are not suitable for representing all kinds of knowledge.
- Scripts are very useful for representing the specific kinds of knowledge for which they were designed.

A script has various components like:

1) **Entry condition**: It must be true before the events described in the script can occur. E.g. in a restaurant script the entry condition must be the customer should be hungry and the customer has money.
2) **Tracks:** It specifies particular position of the script e.g. In a supermarket script the tracks may be cloth gallery, cosmetics gallery etc.
3) **Result:** It must be satisfied or true after the events described in the script have occurred.

   e.g. In a restaurant script the result must be true if the customer is pleased.
   The customer has less money.

4) **Probs:** It describes the inactive or dead participants in the script e.g. In a supermarket script, the probes may be clothes, sticks, doors, tables, bills etc.

5) **Roles:** It specifies the various stages of the script. E.g. In a restaurant script the scenes may be entering, ordering etc.
Now let us look on a movie script description according to the above component.
   a) Script name : Movie
   b) Track : CINEMA HALL
   c) Roles : Customer(c), Ticket seller(TS), Ticket Checker(TC), Snacks Sellers (SS)
   d) Probes : Ticket, snacks, chair, money, Ticket, chart
   e) Entry condition : The customer has money
      The customer has interest to watch movie.

6) **Scenes:**

   a. SCENE-1 (Entering into the cinema hall)

C PTRANS C into the cinema hall
C ATTEND eyes towards the ticket counter C
PTRANS C towards the ticket counters C
ATTEND eyes to the ticket chart
C MBUILD to take which class ticket C
MTRANS TS for ticket
C ATRANS money to TS TS
ATRANS ticket to C

   b. SCENE-2 (Entering into the main ticket check gate)

C PTRANS C into the queue of the gate C
ATRANS ticket to TC
TC ATTEND eyes onto the ticket
TC MBUILD to give permission to C for entering into the hall TC
ATRANS ticket to C
C PTRANS C into the picture hall.

   c. **SCENE-3 (Entering into the picture hall)**

C ATTEND eyes into the chair
TC SPEAK where to sit
C PTRANS C towards the sitting position
C ATTEND eyes onto the screen

   d. SCENE-4 (Ordering snacks)

C MTRANS SS for snacks
SS ATRANS snacks to C
C ATRANS money to SS
C INGEST snacks

   e. SCENE-5 (Exit)

C ATTEND eyes onto the screen till the end of picture
C MBUILD when to go out of the hall
C PTRANS C out of the hall

7) Result:

The customer is happy
The customer has less money

2) TRACKS	:	Ent specialist
3) ROLES	:	Attendant (A), Nurse(N), Chemist (C), Gatekeeper(G), Counter clerk(CC), Receptionist(R), Patient(P), Ent specialist Doctor (D), Medicine Seller (M).
4) PROBES	:	Money, Prescription, Medicine, Sitting chair, Doctor's table, Thermometer, Stetho scope, writing pad, pen, torch, stature.

5) ENTRY CONDITION: The patient need consultation.

Doctor's visiting time on.

6) SCENES:

a. SCENE-1 (Entering into the hospital)

P PTRANS P into hospital

P ATTEND eyes towards ENT department
P PTRANS P into ENT department
P PTRANS P towards the sitting chair

b. SCENE-2 (Entering into the Doctor's Room)

P PTRANS P into doctor's room
P MTRANS P about the diseases
P SPEAK D about the disease
D MTRANS P for blood test, urine test
D ATRANS prescription to P
P PTRANS prescription to P.

P PTRANS P for blood and urine test

c. SCENE-3 (Entering into the Test Lab)

P PTRANS P into the test room

P ATRANS blood sample at collection room
P ATRANS urine sample at collection room
P ATRANS the examination reports

d. SCENE-4 (Entering to the Doctor's room with Test reports)

P ATRANS the report to D

D ATTEND eyes into the report
D MBUILD to give the medicines
D SPEAK details about the medicine to P

P ATRANS doctor's fee
P PTRANS from doctor's room

e.

f. **SCENE-5 (Entering towards medicine shop)**

P PTRANS P towards medicine counter
P ATRANS Prescription to M
M ATTEND eyes into the prescription
M MBUILD which medicine to give
M ATRANS medicines to P
P ATRANS money to M

P PTRANS P from the medicine shop

7) RESULT:

The patient has less money

Patient has prescription and medicine.

Advantages And Disadvantages Of Different Knowledge Representation

Sl. No.	Scheme	Advantages	Disadvantages
1	Production rules	• Simple syntax • Easy to understand • Simple interpreter • Highly Modular • Easy to add or modify	• Hard to follow Hierarchies • Inefficient for large systems • Poor at representing structured descriptive knowledge.
2	Semantic	• Easy to follow hierarchy • Easy to trace associations • Flexible	• Meaning attached to nodes might be ambiguous • Exception handling is difficult • Difficult to program
3	Frame	• Expressive Power • Easy to set up slots for new properties and relations • Easy to create specialized	• Difficult to program • Difficult for inference • Lack of inexpensive software

		Procedures	
4	Script	• Ability to predict events • A single coherent interpretation may be build up from a collection of observations	• Less general than frames • May not be suitable to represent all kinds of knowledge
5	Formal Logic	• Facts asserted independently of use • Assurance that only valid consequence are asserted • Completeness	• Separation of representation and processing • Inefficient with large data sets • Very slow with large knowledge bases