Chapter-5: Structured Knowledge Representation

- Knowledge Representation is an area of AI whose fundamental goal is to represent knowledge in a manner that facilitates inference i.e. drawing conclusion for knowledge.
- It analyzes how to think formally, how to use symbols to represent a domain of discourse along with functions that allow inference about the objects.
- Knowledge representation helps to address the problem like:
  . How do we represent facts about the world?
  . How do we reason about them?
  . What representations are appropriate for dealing with the real world?
  . How to express knowledge in computer understandable form so that reasoning agents can perform well?

Knowledge Mappings

![Knowledge Mappings Diagram]

**Fig: Mappings between Facts and Representations**

Approaches to Knowledge Representation
A good system for knowledge representation should have

i. **Representational Adequacy:**
Ability to represent all kind of knowledge that are needed in the domain.

ii. **Inferential Adequacy:**
Ability to manipulate the representational structure in such a way as to derive new structures corresponding to new knowledge inferred from old.

iii. **Inferential Efficiency:**
Ability to incorporate into the knowledge structure additional information that can be used to focus the attention of the inference mechanism in the most efficient directions.

iv. **Acquisitional Efficiency:**
Ability to acquire new information easily.

**Knowledge Types:**
- **a. Simple relational knowledge:**
The simplest way to represent declarative facts is as a set of relations of the same sort used in database system.
- **b. Inheritable Knowledge**
  Structure must be designed to corresponding to the inference mechanism that are desired.
- **c. Inferential Knowledge**
  Represents knowledge as formal logic. Based on reasoning from facts or from other inferential knowledge. Useless unless there is also as inference procedure that can exploit it.
- **d. Procedural Knowledge (Imperative Knowledge)**
  Knowledge exercised in the performance of some task and processed by an intelligent agent.

**Issues in Knowledge Representation**
- Are any attributes of objects so basic that they have been occurred in almost every problem domain?
- Are there any important relationships that exist among attributes of objects?
- At what level should knowledge be represented?
- How should sets of objects be represented?
- How can relevant part be accessed when they are needed?

**Knowledge can be viewed as different level.**

i. **Knowledge Level:** The most abstract level that describe agent by saying what it knows. Eg: An intelligent taxi might know that the Bagmati bridge connects Thapathali with Kupondole.
ii. **Logical Level:** The level at which the knowledge is encoded into formal sentences. 
   Eg: Links (Bagmati Bridge, Thapathali, Kupondole)

iii. **Implementation Level:** Physical Representation of the sentences in the logical level. 
    Eg: Objets, Dam, Piller etc.

**Knowledge Model**
- A model is a world in which a sentence is true under a particular interpretation.
- There can be several models at once that have the same interpretation.

**Types:**

1. **First Order Predicate Logic**
   - This consists of objects, predicates on objects, connectives and quantifiers
   - Predicates are the relations between objects or properties of the objects.
   - Connectives and quantifiers allows for universal sentences.
   - Relation between objects can be true or false.

2. **Procedural Representation Model**
   - This model of knowledge representation encodes facts along with the sequence of operations for manipulation and processing of the facts.
   - Expert systems are based on this model.
   - It works best when expert follows set of procedures for problem solving. Eg: A medical diagnosis system.

3. **Relational Representation Model**
   - Collections of knowledge are stored in tabular form.
   - Mostly used in commercial databases, relational databases.
   - The information is manipulated with relational calculus using a language such as SQL, Oracle etc.
   - This is flexible way to store information but not good for storing complex relationships.
   - Problem arises when more than one subject area is attempted.
   - A new knowledgebase from scratch has to be built for each area of expertise.

4. **Hierarchical Representation Model**
   - Based on inherited knowledge and relationship and shared attributes between objects.

5. **Semantic Nets**
   - Semantic network is an alternative to predicate logic as a form of knowledge representation.
   - The structure of a semantic net is shown graphically in terms of nodes and the arcs connecting them.
   - Nodes are sometimes referred to as objects.
   - Arcs represent the links or edges.
   - The links are used to express relationships.
   - Two types of commonly used links are i. IS-A, and ii. A-KIND-OF
   - IS-A means “is an instance of” and refers to a specific member of a class.
- The link A-KIND-OF is used here to relate one class to another.
- Nodes are to represent physical objects, concepts, or situation.
- The idea is that we can store our knowledge in the form of a graph, with nodes representing objects in the world, and arcs representing relationships between those objects.
- Semantic network is a declarative graphic representation that can be used to represent knowledge and support automated systems for reasoning about the knowledge.
- For example, the following:

```
Semantic Network is intended to represent the data:
- Tom is a cat.
- Tom caught a bird.
- Tom is owned by John.
- Tom is ginger in colour.
- Cats like cream.
- The cat sat on the mat.
- A cat is a mammal.
- A bird is an animal.
- All mammals are animals.
- Mammals have fur.
- It is argued that this form of representation is closer to the way human’s structure knowledge.
```
Frames (slot and filler structure)
- Frame is a type of schema used in many AI applications including vision and natural language processing that provides a convenient structure for representing.
- Frame is similar to a record structure and corresponding to the fields and values are slots and slot fillers.
- Basically it is a group of slots and fillers that defines a stereotypical object.
- A single frame is not much useful. Frame systems usually have collection of frames connected to each other.
- Value of an attribute of one frame may be another frame.
- Frames are also useful for representing commonsense knowledge.
- While semantic nets are basically a two-dimensional representation of knowledge, frames add a third dimension by allowing nodes to have structures.
- By using frames in the filler slots and inheritance, very powerful knowledge representation systems can be built.
- Frame-based expert systems are very useful for representing causal knowledge because their information is organized by cause and effect.
- Frames are generally designed to represent either generic or specific knowledge.
- A frame for a book is given below.

<table>
<thead>
<tr>
<th>Slots</th>
<th>Fillers</th>
</tr>
</thead>
<tbody>
<tr>
<td>publisher</td>
<td>Thomson</td>
</tr>
<tr>
<td>title</td>
<td>Expert Systems</td>
</tr>
<tr>
<td>author</td>
<td>Giarratano</td>
</tr>
<tr>
<td>edition</td>
<td>Third</td>
</tr>
<tr>
<td>year</td>
<td>1998</td>
</tr>
<tr>
<td>pages</td>
<td>600</td>
</tr>
</tbody>
</table>

Conceptual Dependency (CD) (Strong slot and filler structure)
- Conceptual Dependency originally developed to represent knowledge acquired from natural language input.
- The goals of this theory are:
  - To help in the drawing of inference from sentences.
  - To be independent of the words used in the original input.
- For any 2 (or more) sentences that are identical in meaning there should be only one representation of that meaning.
- It has been used by many programs that portend to understand English.
- CD provides:
• a structure into which nodes representing information can be placed

• a specific set of primitives

• at a given level of granularity.

- Sentences are represented as a series of diagrams depicting actions using both abstract and real physical situations.
- The agent and the objects are represented.
- The actions are built up from a set of primitive acts which can be modified by tense.

**Examples of Primitive Acts are:**

**ATRANS**  
-- Transfer of an abstract relationship. *e.g. give.*

**PTRANS**  
-- Transfer of the physical location of an object. *e.g. go.*

**PROPEL**  
-- Application of a physical force to an object. *e.g. push.*

**MTRANS**  
-- Transfer of mental information. *e.g. tell.*

**MBUILD**  
-- Construct new information from old. *e.g. decide.*

**SPEAK**  
-- Utter a sound. *e.g. say.*

**ATTEND**  
-- Focus a sense on a stimulus. *e.g. listen, watch.*

**MOVE**  
-- Movement of a body part by owner. *e.g. punch, kick.*

**GRASP**  
-- Actor grasping an object. *e.g. clutch.*

**INGEST**  
-- Actor ingesting an object. *e.g. eat.*

**EXPEL**  
-- Actor getting rid of an object from body. *e.g. throw*

Example:

*John gave Mary a book.*

*Mary took a book from John.*
**SCRIPT**

A script is a remembered precedent, consisting of tightly coupled, expectation-suggesting primitive-action and state change frames [Winston, 1992].

A script is a structured representation describing a stereotyped sequence of events in a particular context [Luger, Stubblefield, 1998, p.324].

That is, extend frames by explicitly representing expectations of actions and state changes. Find primitives to describe the world like PTRANS for “transfer physical location of an object (= go)” and ATRANS for “transfer a relationship (= give)”.

**A Restaurant Script**

<table>
<thead>
<tr>
<th>Script: RESTAURANT</th>
<th>Track: Coffee Shop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Props: Tables</td>
<td>Entry cond.: S hungry</td>
</tr>
<tr>
<td>Menu</td>
<td>S has money</td>
</tr>
<tr>
<td>F=Food</td>
<td>Results: S has less money</td>
</tr>
<tr>
<td>Check</td>
<td>O has more money</td>
</tr>
<tr>
<td>Money</td>
<td>S is not hungry</td>
</tr>
<tr>
<td>Roles S=Customer</td>
<td>S is pleased (optional)</td>
</tr>
<tr>
<td>W=Waiter</td>
<td>M=Cashier</td>
</tr>
<tr>
<td>C=Cook</td>
<td>O=Owner</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scene 1: Entering</th>
<th>Scene 2: Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>S PTRANS S into restaurant</td>
<td>(Menu on table) (W brings menu)</td>
</tr>
<tr>
<td>S ATTEND eyes to tables</td>
<td>S PTRANS menu to S</td>
</tr>
<tr>
<td>S MBUILD where to sit</td>
<td>S MTRANS food list to S</td>
</tr>
<tr>
<td>S PTRANS S to table</td>
<td>S MTRANS signal to W</td>
</tr>
<tr>
<td>S MOVE S to sitting position</td>
<td>W PTRANS W to table</td>
</tr>
<tr>
<td></td>
<td>S MTRANS ‘I want F’ to W</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scene 3: Eating</th>
<th>Scene 4: Exiting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C ATRANS F to W</td>
<td>W MOVE (write check)</td>
</tr>
<tr>
<td>W ATRANS F to S</td>
<td>W PTRANS W to S</td>
</tr>
<tr>
<td>S INGEST F</td>
<td>W ATRANS check to S</td>
</tr>
<tr>
<td>(Option: Return to Scene 2 to order more; otherwise, go to Scene 4)</td>
<td>S ATRANS tip to S</td>
</tr>
<tr>
<td></td>
<td>S PTRANS S to M</td>
</tr>
<tr>
<td></td>
<td>S ATRANS money to M</td>
</tr>
</tbody>
</table>
S PTRANS S to out of restaurant