

# Selecting Integrated Approach for Knowledge Representation by Comparative Study of Knowledge Representation Schemes

Pushpa Kesarwani\*, Prof. A.K. Misra\*\*

Computer Science & Engineering Department  
MNNIT, Allahabad  
Allahabad, India

\*Email: Pushpak28@yahoo.com

\*\*Email: akm@mnnit.ac.in

**Abstract-** In Artificial Intelligence, knowledge representation is a combination of data structures and interpretive procedures that leads to knowledgeable behavior. Therefore, it is required to investigate such knowledge representation technique in which knowledge can be easily and efficiently represented in computer. This research paper compares various knowledge representation techniques and proves that integrated approach is a more efficient and more accurate knowledge representation scheme.

**Index Terms-** Knowledge Representation, Predicate Logic, Frames, Script, Production Rule, Semantic Network.

## I. INTRODUCTION

A knowledge representation (KR) is an idea to enable an individual to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it [2]. The knowledge acquired from experts or induced from a set of data must be represented in a format that is both understandable by humans and executable on computers. For Good Knowledge Representation Languages, there should some qualities:

1. Expressive
2. Concise
3. Unambiguous
4. Independent of context
  - What you say today will still be interpretable tomorrow.
5. Efficient
  - The knowledge can be represented in a format that is suitable for computers.
  - Practical inference procedures exist for the chosen format.
6. Effective
  - There is an inference procedure which can act on it to make new sentences.

## II. KNOWLEDGE REPRESENTATION TECHNIQUES

### 2.1 KNOWLEDGE REPRESENTATION USING LOGIC

Logic is a formal language for representing facts in a precise and unambiguous way [4]. There two basic forms of computational logic are **propositional logic** and **predicate logic**. A proposition is nothing more than a statement that is either true or false. In **propositional logic**, we use symbols, such as letters of the alphabet, to represent propositions, premises, or conclusions. For example,

The mail carrier comes Monday through Friday.  $\rightarrow$  A (premise)

Today is Sunday.  $\rightarrow$  B (premise)

The mail carrier will not come today.  $\rightarrow$  C (conclusion)

To form more complex premises, two or more propositions can be combined, using logical connectives such as AND, OR, NOT, IMPLIES, and EQUIVALENT. It can be manipulated using the rules of propositional logic to infer new conclusions. Because propositional logic deals primarily with complete statements and whether they are true or false; its ability to represent real-world knowledge is limited.

**Predicate logic** is useful for showing logic relationships and their reasoning. In predicate logic statements can be split into words. Predicate logic employs the notions of constant, variable, function, predicate, logical connectives and quantifiers to represent facts [7]. A predicate with one argument expresses a property of an object for e.g.

Student(smith). A predicate with two or more arguments expresses a relation between objects for e.g. likes (smith, Mary). Predicate with no arguments is just simple proposition logic [2].

### 2.2 KNOWLEDGE REPRESENTATION USING SEMANTIC NET

Semantic networks are an alternative to predicate logic as a form of knowledge representation [1]. Semantic networks focus on the relationships between different concepts. They are graphical depictions of knowledge composed of nodes and links (arcs). Nodes are represented by objects, and arcs representing relationships between those objects. An object can be any physical item, such as a book, a car, a desk, or even a person. Nodes can also be concepts, events, or actions. Some of the most common arcs are of the IS-A or HAS-A type. IS-A is used to show a class relationship (i.e., that an object belongs to a larger class or category of objects). HAS-A links are used to identify characteristics or attributes of object nodes. But the representation of semantics networks in a computer is very complex. As only objects and there associations are stored in

computer, therefore retrieving the actual and correct information from semantics networks is a very difficult task [3].

### 2.3 KNOWLEDGE REPRESENTATION USING FRAME:

A **frame** is a data structure that includes all the knowledge about a particular object. This knowledge is organized in a special hierarchical structure that permits a diagnosis of knowledge independence. Frames are basically an application of object-oriented programming for artificial intelligence and Expert System. Frames provide a concise structural representation of knowledge in a natural manner. The knowledge in a frame is partitioned into slots. A slot can describe declarative knowledge (e.g., the color of a car) or procedural knowledge. A frame includes two basic elements: slots and facets. A **slot** is a set of attributes that describe the object represented by the frame. Each slot contains one or more facets. The **facets** (subslots) describe some knowledge or procedural information about the attribute in the slot. Most artificial intelligence systems use a collection of frames linked together in a certain manner to show their relationship. This is called a *hierarchy of frames*. The hierarchical arrangement of frames permits inheritance frames.

### 2.4 KNOWLEDGE REPRESENTATION USING SCRIPT

A script is a term proposed by Schank, and it refers to a form of knowledge representation [8]. A script is a structured representation describing a stereotyped sequence of events in a particular context [5]. For Example, when we go to a restaurant, we usually 'enter the restaurant', 'wait', 'sit down', 'get the menu and decide what to eat', 'order the dish', 'wait until the dish has come', and so on. This sequence can be said to be script knowledge in the situation of 'eating at a restaurant'[6].

### 2.5 KNOWLEDGE REPRESENTATION USING PRODUCTION RULES

**Production rules** are the most popular form of knowledge representation for expert system and automated decision support (ADS) systems. Knowledge is represented in the form of condition/action pairs: IF this condition (premise) occurs, THEN some action (result or conclusion) will occur. Consider examples, "If the stop light is red AND you have stopped, THEN a right turn is okay."

Two types of rules are common in artificial intelligence: knowledge and inference rules. Knowledge rules (declarative rules) state all the facts and relationships about a problem. **Inference rules** (procedural **rules**) advice on how to solve a problem, given that certain facts are known. The knowledge engineer separates the two types of rules: Knowledge rules go to the knowledge base, whereas inference rules become part of the inference engine.

Example of the knowledge rules:

Rule 1: IF an international conflict begins, THEN the price of gold goes up.

Rule 2: IF the inflation rate declines, THEN the price of gold goes down.

Rule 3: IF the international conflict lasts more than seven days and IF it is in the Middle East, THEN buy gold.

Inferences rules contain rules about rules and thus are also called **metarules**. They pertain to other rules (or even to themselves). Inference (procedural) rules may look like this:

Rule 1: IF the data needed are not in the system, THEN request them from the user.

Rule 2: IF more than one rule applies, and THEN deactivate any rules that add no new data.

### III. SELECTION OF REPRESENTATION METHODS

Knowledge Representation methods all have advantages and limitations. Production rules are popular in the design of first-generation expert system. The object-oriented method has become very popular in recent years. Predicate logic provides a theoretical foundation for rule-based inferences. To navigate the problem associated with single knowledge representation technique the integrated knowledge representation came in picture.

Sometimes, no single knowledge representation method is by itself ideally suited for all tasks. When several sources of knowledge are used simultaneously, the goal of uniformity may have to be sacrificed in favor of exploiting the benefits of multiple knowledge representations, each tailored to a different subtask. The necessity of translating among knowledge representations becomes a problem in these cases. Nevertheless, some recent expert system shells use two or more knowledge representation Schemes, e.g., the CORVID, KRYPTON, MANTRA, FRORL system.

Production rules and frames comprise a successful combination of knowledge representation methods. Production rules do not provide a totally effective Representation facility for many ES applications. But their expressive power is inadequate for defining terms and for describing domain objects and static relationships among objects. Frames are objects and fit the paradigm of an object-oriented approach to ES development well. Objects encapsulate properties and Actions, just as frames store knowledge. Frames are organized into classes, which are organized into hierarchies. Each frame inherits its properties from its parent frame. Rules can either guide the frames behaviors or be embedded in the frame. The object-oriented paradigm fits the hybrid ES structure well in working with frames and rules. CORVID and many other development tools have adopted this integrated approach.

**FRORL** -It uses frames and production rules.

**MANTRA** – It uses first order logic, semantic networks, production system and terminological language.

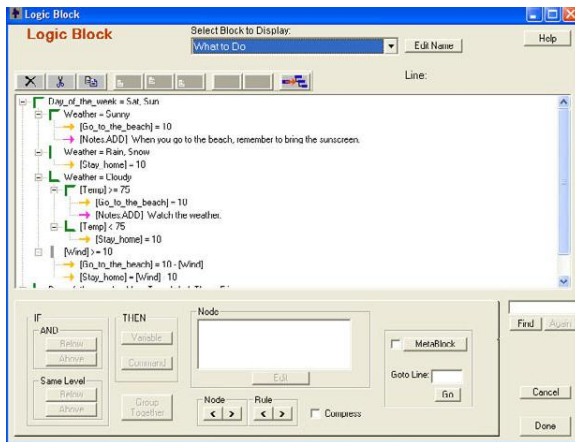
**KRYPTON** – It uses frames and predicate logic.

**CORVID** - It uses frames, production rules and object – oriented language.

Expert systems are computer programs that Provide problem solving knowledge and Situation-specific advice comparable to talking to a human expert. CORVID is a Knowledge Automation Expert Systems tool. Exsys Corvid provides ways to capture expert knowledge and decision-making processes that allow computers to emulate the interaction people have with human experts to solve problems. Corvid systems are portable across platforms for enterprise-wide integration and run from Web sites, intranets and client-side. In Exsys Corvid, rules are written in English and Algebra. They are easy to read and very similar to

what the expert would tell you if you asked, “How did you make that decision?” Exsys Corvid is designed to help developers (domain experts) build rules that are easy to read, understand and maintain.

Exsys Corvid has a unique way to define, organize and structure rules into logically related modules. A Logic Block is made up of one or more structured logic diagrams.



#### IV. CONCLUSION

This paper has presented the comparison between five representation schemes shown in annexure 1 and the objective is to analyse the power and expressiveness of a system. Each knowledge representation schemes has advantages and disadvantages. Combination of two or more representation scheme may be used to for making the system more efficient and improving the knowledge representation.

#### REFERENCES

- [1] Tilotma Sharma, Deepali Kelkar, "A Tour Towards knowledge Representation Techniques", International Journal of Computer technology and Electrical Engineering(IJCTEE), Vol. 02, issue 2, 2011, 2249-6343
- [2] Poonam Tanwar, Dr. T.V. Prasad, Dr. Mahendra S. Aswal, "Comparative Study of Three Declarative Knowledge Representation Techniques", International Journal on Computer Science and Engineering, Vol. 02, No. 07, 2010, 2274-2281
- [3] Amjad Ali, Mohammad Abid Khan, "Selecting Predicate Logic for Knowledge Representation by Comparative Study of Knowledge Representation Schemes", International conference on Emerging Technologies, 2009 IEEE
- [4] Knowledge representation, Retrieved on May 05, 2007, from <http://www.informatics.susx.ac.uk/books/computers-and-thought/chap3>,
- [5] M. Kerber, "Knowledge Representation I", Introduction to AI 06-08775, 2004/05.
- [6] T. Fujiki, H. Nanba, and M. Okumura, "Automatic acquisition of script knowledge from a text collection", EACL 2003 in Budapest, Hungary, 91-94.
- [7] Mylopoulos, J., "An overview of knowledge representation", 1981, ACM press New York, NY, USA.
- [8] C. Roger, Schank and Robert, P. Abelson, "Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures", Lawrence Erlbaum Associates, 1977.

| SNO. | Knowledge structure    | Logic   | Semantic Net  | Frame   | Script   | Production Rule  |
|------|------------------------|---|---|---|--|--|
| 1    | Merits                 | <ul style="list-style-type: none"> <li>* Facts asserted independently of use</li> <li>* Assurance that only valid consequences are asserted.</li> <li>* Its providing a better way to do reasoning by providing a way of deducing new from old one.</li> <li>* It can be used for proving the statements.</li> <li>* Quantified and existential statements are easily represented.</li> </ul> | <ul style="list-style-type: none"> <li>* Easy to follow hierarchy.</li> <li>* Easy to trace associations.</li> <li>* Easy to visualize and understand.</li> <li>* It is flexible</li> </ul> | <ul style="list-style-type: none"> <li>* Expressive power</li> <li>* Easy to set up slots for new properties and relations.</li> <li>* Easy to create specialized procedures.</li> <li>* Easy to include default information and easy detect missing values</li> <li>* Very flexible</li> </ul> | <ul style="list-style-type: none"> <li>* It provide a high degree of flexibility</li> <li>* Ability to predict events</li> <li>* A single coherent interpretation may be build up from a collection of observations.</li> </ul>  | <ul style="list-style-type: none"> <li>* Simple syntax</li> <li>* Easy to understand</li> <li>* Simple interpreter</li> <li>* Highly modular</li> <li>* It is flexible(to add or modify)</li> <li>*Naturalness of expression</li> </ul>          |
| 2    | Demerits               | <ul style="list-style-type: none"> <li>* Not possible to store and represent infinite sentences.</li> <li>* Processing is not efficient and is time consuming.</li> <li>* Less expressive</li> <li>* Used for representing statics facts only.</li> <li>* Inefficient with large data sets.</li> <li>* Very slow with large knowledge bases.</li> </ul>                                       | <ul style="list-style-type: none"> <li>* Meaning attached to nodes might be ambiguous.</li> <li>* Exception handling is difficult</li> <li>* Difficult to program</li> </ul>                | <ul style="list-style-type: none"> <li>* Difficult to program</li> <li>* Difficult for inference</li> <li>* Lack of inexpensive software</li> </ul>   | <ul style="list-style-type: none"> <li>* Complex concepts, influences and dependencies are difficult to represent.</li> <li>* The precise structure is hard to select to achieve an optimal performance.</li> <li>* Less general than frames.</li> <li>* May not be suitable to represent all kinds of knowledge.</li> </ul> | <ul style="list-style-type: none"> <li>* Hard to follow hierarchies.</li> <li>* Inefficient for large systems</li> <li>* Not all knowledge can be expressed as rules</li> <li>* Poor at representing structured descriptive knowledge</li> </ul> |
| 3    | Nearest data structure | Rule based system   | Graph   | Class in object oriented programming  | Class of events  | <ul style="list-style-type: none"> <li>*Tree-structure</li> <li>*Divide and conquer,</li> </ul>  |
| 4    | Searching algorithm    | <ul style="list-style-type: none"> <li>*Breadth First</li> <li>*Depth first</li> <li>*Forward Chaining</li> <li>*Backward Chaining</li> </ul>   | <ul style="list-style-type: none"> <li>* Intersection Search</li> <li>* Inheritance</li> <li>* Breadth First</li> <li>* Depth First</li> <li>* Heuristic Search</li> </ul>                  | <ul style="list-style-type: none"> <li>* Inheritance</li> <li>* frame matching</li> </ul>   | <ul style="list-style-type: none"> <li>* Inheritance</li> <li>*Tree-Structure</li> </ul>   | <ul style="list-style-type: none"> <li>*pattern matching algorithm(Rete algorithm)</li> <li>*forward chaining and backward chaining</li> </ul>   |

