ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (619-624), Month: April - June 2015, Available at: www.researchpublish.com

Knowledge Representation by Using Slot and Filler Structures

Navaneetha A R[1], Sunita Kanadikar[2] , Mahesh Kaluti[3]

Visvesvaraya Technological University, ALVAS Institute of Engineering and Technology, Mangalore, India

***Abstract*: In order to have intelligent computers, or robots, they will need to be able to understand the world, and to understand it, they must have knowledge of it. One need both the large amount of knowledge and some mechanism for manipulating that knowledge to create solution to problem. The knowledge representation is used to solve these problems encountered in artificial intelligence. A variety of ways of representing knowledge have been exploited. Many issues need to be considered when deciding on the representation scheme for knowledge. Hence the concept of a "slot and filler structure" is used to represent knowledge. In this paper we will be looking at different types of slot and filler representations starting with weak slot and filler structures and moving onto stronger slot and filler structures based representations.**

***Keywords:* knowledge representation, slot and filler structure, weak slot and filler structures, stronger slot and filler structures based representations.**

1. INTRODUCTION

Programming a computer to act like a human is a difficult task and requires that the computer system be able to understand and process commands in natural language, store knowledge, retrieve and process that knowledge in order to derive conclusions and make decisions, learn to adapt to new situations, perceive objects through computer vision and have robotic capabilities to move and manipulate objects. A robust AI system must be able to store knowledge, apply that knowledge to the solution of problems and acquire new knowledge through experience[1]. Knowledge is a general term. Knowledge and representation are distinct entities that play central but distinguishable roles in the intelligence system. Knowledge is a description of the world. It determines a system's competence by what it knows. Representation is the way knowledge is encoded. It defines the performance of a system in doing something[4]. Different types of knowledge require different kinds of representation.

The knowledge in slot and filler systems consists of structures as a set of entities and their attributes. Slot is an attribute value pair in its simplest form. Filler is a value that a slot can take could be a numeric, string value or a pointer to another slot. The structure enables attribute values to be retrieved quickly and Assertions are indexed by the entities[6]. Properties of relations are easy to describe. It allows ease of consideration as it embraces aspects of object oriented programming. This structure is called a weak slot and filler structure. But later we used strong slot and filler structures. It represent links between objects according to more rigid rules. Specific notions of what types of object and relations between them are provided. Represent knowledge about common situations than weak slot and filler structure.

2. OVERVIEW OF KNOWLEDGE REPRESENTATION

Knowledge is a general term. Knowledge is a progression that starts with data which is of limited utility. By organizing or analyzing the data, we understand what the data means, and this becomes information. The interpretation or evaluation of information yield knowledge. An understanding of the principles within the knowledge is wisdom [10].

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (619-624), Month: April - June 2015, Available at: www.researchpublish.com

Let's look at various kinds of knowledge that might need to represent AI systems. *Objects*-Objects are defined through facts. *Events*-Events are actions. *Performance*-Playing the guitar involves the behavior of the knowledge about how to do things. *Meta-knowledge*-Knowledge about what we know. *Facts*- they are truths about the real world on what we represent. This can be considered as knowledge level.

3. WEAK SLOT AND FILLER STRUCTURES

Weak slot and filler structures are a data structure and the reasons to use this data structure are as follows: It enables attribute values to be retrieved quickly. Assertions are indexed by the entities. Binary predicates are indexed by first argument. Properties of relations are easy to describe. And it allows ease of consideration as it embraces aspects of object oriented programming. We describe the two kinds of structures here: *Semantics* and *Frames.*

3.1 Semantic Net:

Semantic nets are a simple way of representing the relationships between entities and concepts. The major idea is that find the meaning of a concept comes from its relationship to other concepts, and that, the information is stored by interconnecting nodes with labeled arcs.

3.1.1 Representation In A Semantic Net:

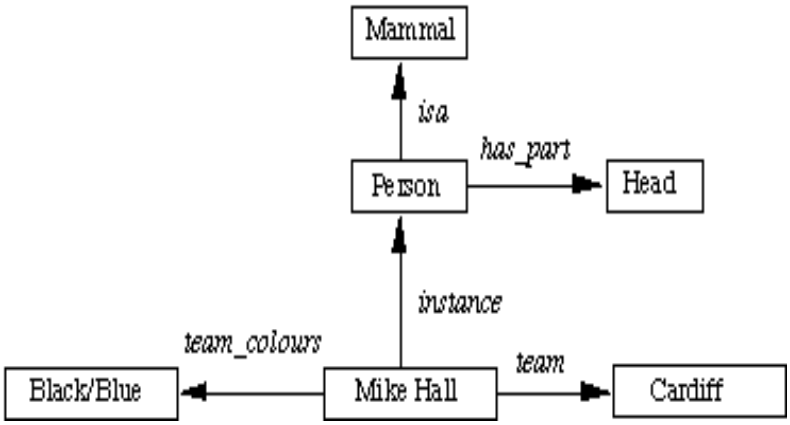


Fig 1: The physical attributes of a person in semantic network

These values can also be represented in logic as: isa(person, mammal), instance(Mike-Hall, person) team(Mike-Hall, Cardiff).

3.2 Frames:

Frames can also be regarded as an extension to Semantic nets. Indeed it is not clear where the distinction between a semantic net and a frame ends. Initially, semantic nets were used to represent labeled connections between objects. As tasks became more complex, the representation needs to be more structured. The more structured the system, the more beneficial to use frames. A frame is a collection of attributes or slots and associated values that describe some real-world entity. Frames on their own are not particularly helpful but frame systems are a powerful way of encoding information to support reasoning. Frames can inherit slots from parent frames. For example, man might inherit properties from Ape or Mammal (A parent class of man).

3.2.1 Frame Knowledge Representation:

Consider the example first discussed in Semantic Nets:

Person

Isa: Mammal

Cardinality:

Adult-Male

Isa: Person

Cardinality:

Cricket-Player

Isa: Adult-Male

Cardinality:

Height:

Weight:

Position:

Team

Team-Color: Back

Isa: Cricket-Player

Cardinality:

Tries:

Dhoni

Instance: Back

Height: 6-0

Position: Centre

Team: India

Team-Colours: Black

Cricket-Team

Isa: Team

Cardinality:

Team-size: 10

Coach:

Cricket team":

Instance: Indian Team

Team size: 10

Coach: Amarnadh

Players:{Sachin, Yuvaraj Singh, Srinath, Harbajan…..}

Here the frames Person, Adult-Male, Cricket-Player and Team are all classes. And the frames dhoni and cricket team are instances.

4. STRONG SLOT AND FILLER STRUCTURES

The main problem with semantic networks and frames is that they lack formality, there is no specific guideline on how to use the representation and if I use the word "has" in a way other than "physical property", your reasoning might break down and isa and instance attributes seem clearly defined, but the attributes may not be. In frame when things change, we need to modify all frames that are relevant – this can be time consuming.

Strong slot and filler structures typically represent links between objects according to more rigid rules, specific notions of what types of object and relations between them are provided. We have types of strong slot and filler structures:

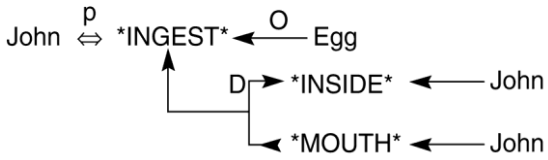
1. Conceptual Dependency (CD)

2. Scripts

3. Cyc

4.1 Conceptual Dependency:

Conceptual dependency (CD) slot and filler structures used to represent the kind of knowledge about events that is usually conveyed in natural language sentences. Goal is to represent knowledge so as to Facilitate drawing inferences from sentences. Be independent of the language in which the sentences were originally stated. Conceptual dependency has two Claim. First For any two sentences that are identical in meaning, regardless of language, there should be only one representation. Second any information in a sentence that is implicit must be made explicit in the representation of the meaning of that sentence. Example is shown below



The sentence is "John ate the egg". The INGEST act means to ingest an object (eat, drink, swallow). The P above the double arrow indicates past test. The INGEST action must have an object (the O indicates it was the object Egg) and a direction (the object went from John's mouth to John's insides). We might infer that it was "an egg" instead of "the egg" as there is nothing specific to indicate which egg was eaten. We might also infer that John swallowed the egg whole as there is nothing to indicate that John chewed the egg.

4.1.1Conceptual Dependency primitive:

There are some primitives in CD. *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. By using above primitives we are representing the knowledge.

4.2 Scripts:

Script is a structure that describe a stereotyped sequence of events in a context. Script consist of a set of slots. Each slot may be some information about what kind of values it may contain as well as a default value to be used if no other information is available.The components of a script include: *Entry Conditions* -- these must be satisfied before events in the script can occur. *Results* -- Conditions that will be true after events in script occur. *Props* -- Slots representing objects involved in events. *Roles*-- Persons involved in the events. *Track* --Variations on the script. Different tracks may share components of the same script. *Scenes* -- The sequence of events that occur. *Events* are represented in *conceptual dependency* form. Example- Restaurant Script Structure.

SCRIPT : Restaurant

TRACK : Oberoi

PROPS : Tables, Chairs, Menu, Money, Food

ROLES:Customer, Waiter, Cashier, Owner, Cook

Entry Conditions

(a) Customer is hungry

(b) Customer has money

Scene 1: *Entering*

(a) Customer PTRANS into Restaurant.

(b) Customer ATTEND eyes to the tables.

(c) Customer MBUILD where to sit.

(d) Waiter PTRANS customer to an empty table.

(e) Customer MOVES to sitting position.

Scene 2: ***Ordering***

(a) Waiter PTANS the menu.

(b) Customer MBUILD choice of food.

(c) Customer MTRANS signal to waiter.

(d) Waiter PTRANS to table.

(e) Customer MTRANS I want food to waiter.

Scene 3: Eating

(a) Cook ATRANS food to waiter.

(b) Waiter ATRANS food to customer.

(c) Customer INGESTS food.

(Option: Return to scene 2 to order more: otherwise, go to scene 4)

Scene 4: Eating

(a) Customer MTRANS to waiter.

(b) Waiter PTRANS to customer.

(c) Waiter ATRANS bill to customer.

(d) Customer ATRANS tip to waiter.

(e) Customer PTRANS to cashier.

(f) Customer ATRANS money to cashier.

(g) Customer PTRANS out of restaurant.

Results

(a) Customer is no more hungry.

(b) Customer is satisfied.

(c) Owner gets money.

4.3 CYC:

Cyc is a very large knowledge based project aimed at capturing human commonsense knowledge. Goal of Cyc is to encode the large body of knowledge that is so obvious that it is easy to forget to state it explicitly. Like CD, CYC represent a specific theory of how to describe the world. CD provide a specific theory of representation of events. CYC provide a specific theory of representation of events, objects, attributes and so on. The CYC is coded by using the following methods: by hand. Special CYC languages such as LISP like, Frame Based, Multiple inheritances, Slots are fully fledged objects. Generalized inheritance any link not just 'isa' and instance.

CYC's knowledge is encoded in a representation language called CYCL. CYCL is a frame-based system that incorporates most of the techniques like multiple inheritance, slots as full-fledged objects, transfers-through, mutually-disjoint-with, etc. CYCL generalizes the notion of inheritance so that properties can be inherited along any link, not just isa and instance.

5. CONCLUSION

The knowledge representation problem concerns the mismatch between human and computer 'memory', i.e., how to encode Knowledge. Hence by using slot and filler structure is used to represent the knowledge. Here in this paper we learnt about two kinds of slot and filler structures. Both strong slot and filler structure and weak slot and filler structure are used to represent different facts, events, objects etc. We do not have much experience with the engineering problems of building and maintaining very large knowledge bases. In future, we should have a lot of tools that checks consistency in the knowledge base.

REFERENCES

[1] Randall Davis, Howard Shrobe, and Peter Szolovits. AI Magazine 14(1) Spring, 17-33. "What is a knowledge representation? (1993).

[2] J. R. Anderson. Cognitive Skills and Their Acquisition. Hillsdale, N. J., Lawrence Erlbaum (1981).

[3] G.J. Sussman. A Computational Model of Skill Acquisition. Ph.D. thesis. Cambridge, MA, MIT (1973).

[4] "Whatisa Knowledge Representation?" Association for the Advancement of Artificial Intelligence 14(1)http:// www.aaai.org/ojs/index.php/aimagazine/article/view/1029/947 (1993).

[5] "AITopics/Representation".Associat in For the Advancement of Artificial Intelligence http://www.aaai.org/aitopics/ pmwiki/pmwiki.php/AI Topics/ Representation. Retrieved 23 March.

[6] Rathke, C., "Object-oriented programming and frame-based knowledge representation", 5th International Conference, Boston, (1993).

[7] Knowledge Representation Techniques- Prof. Dr. Shaiq Haq, Mechatronics Engineering, AIR University, http://www.wecuw.edu.pk/downloads/ai/06_AI_Lectureon_Knowledge_Representation_Techniques.pdfIslamabad.

[8] Von Altrock, Constantin. Fuzzy logic and NeuroFuzzy applications explained. Upper Saddle River, NJ: Prentice Hall PTR. ISBN 0-13-368465-2 (1995). ++++http://whatis.techtarget.com/definition/0, ,sid9_gci835674,00.html

[9] http://en.wikipedia.org/wiki/Temporal_logic.

[10] Venema, Yde, "Temporal Logic," in Goble, Lou, ed., The Blackwell Guide to Philosophical Logic. Blackwell (2001).

[11] E. A. Emerson and C. Lei, modalities for model checking: branching time logic strikes back, in Science of Computer Programming 8, p 275-306, (1987).

[12] E.A. Emerson, Temporal and modal logic, Handbook of Theoretical Computer Science, Chapter 16, the MIT Press, (1990).

[13] Han Reichgelt, "Knowledge Representation: An AI Perspective", Chapter 5 (Semantic Networks).