UDC 004.[822+823]

SEMANTIC NETWORKS AND FRAMES: KNOWLEDGE REPRESENTATION PROBLEMS. THOUGHTS ON SEMANTIC FRAMES

ALIAKSANDR DUNCHANKA, DZMITRY PIATKIN
Polotsk State University, Belarus

This article describes two major knowledge representation models – semantic nets and frames, points out their flaws, shows the possibilities to combine both formalisms into a more powerful one, providing this notation with necessary extensions, and recommends how to do it using Lisp and UML as fundamental technologies.

Foreword

Semantic networks and frames are two formalisms, used to capture information about the world and represent knowledge in forms that allow utilizing computer systems to deal with information from different problem domains and solve complex tasks.

Some sources blindly claim both to be powerful and universal solutions for knowledge representation. This isn't just a bold statement, but a completely false one. Each has its own conceptual flaws and practical limitations, which narrow down a range of their possible applications.

Semantic networks

Semantic network (SN) has a form of directed or undirected graph consisting of *vertices*, which denote entities or conceptions, and *edges*, which specify relations between them. The number of all possible relations is finite – less than 300. Any other type of relationship can be represented as their combination.

Semantic nets are good only for one thing – expressing relations. Period. Any attempt to use them for other purposes immediately draws their apparent disadvantages:

1) SN can't describe complex *procedural knowledge*, only very simple – linear – algorithms. In fact, there are much better notations for this task out there – *activity diagrams* and *flow-chart diagrams*;

2) SN can express only binary relations, which means a lot of redundancy;

3) even when it comes to relations, *entity-relationship diagram* is a much more intuitive and expressive tool (especially for depicting complex structures);

4) there is no standard definition of link names;

5) it's impossible to describe all aspects of problem domain without using special extensions (like qualifiers);

6) loads of vertices with the same name make it hard to search for a specific node.

Frames

Frames were originally derived from semantic networks and are a part of structure-based knowledge representations. The basic idea behind frames relates to the human psychology: people tend to see real world as a set of composite objects, where each one can be classified as a member of a concrete group (type).

A *frame* is a minimally possible description of some object, event, situation or process – a part of a knowledge base, which can be analyzed independently from other fragments. Frames have solid structure and are composed of standard elements called *slots*. Every slot has a name and contains a value.

While the frame model is a major improvement over semantic networks, it closely based on the concepts from *object-oriented programming* (OOP) paradigm and, therefore, inherits of lot of its problems:

1) There is a huge difference between "real" objects and "knowledge" about them, so any class, which encapsulates the information about some entity, can't be simply modeled after it. Here's an example: suppose we have two real people – a male and a female. To produce a new human both need to have an intercourse and give birth to a baby. However, when we deal (on the "knowledge level") with two frames, which represent those people (*Mal*e and *Female* – both subclasses of *HumanBeing)*, there is no need to describe any special procedures (like "*make_ove()*" and "*give_birth()*") to make a new frame. Actually, nothing stops the processing system (whether it's a natural mind or artificial intelligence) from involving into the creative cognitive process – imagination – and generating fake knowledge, e.g. creating an instance of human frame without a real counterpart in existence.

2) Incorrect or incomplete information leads to shallow knowledge. Perceptual blindness and other factors (like expectations, emotional state) can have significant impact on the whole perception process and result in frames with empty or ambiguous slots. For example, when a perceiver is being exposed to something for the first time or for a short period, its consciousness creates a frame for this distal stimulus, but this frame may be not fully-constructed, because all properties of the stimulus are not known yet.

3) Uncertainty and difficulties with classification. Let's imagine a man driving a car. It is evident that technically the driver is not a component of the car itself, so he can't be a member of the *Car* frame. At the same

time when we view this system as a physical object that moves in space it can be represented as a single entity: man's and car's masses sum and, along with the speed, define how much kinetic energy this object possesses. In this case both CAN be organized as some virtual composite frame, containing all parts of the system.

4) It's difficult (and sometimes impossible) to make changes in the core hierarchy and hard to handle exceptions.

5) Different pieces of information, combined in a single frame, can't be represented as a sequence of definitions.

6) There is no way to control the inference process. *Attached procedures* is a lousy way.

7) In OOP there are three techniques used to express behavioral aspect of classes (procedural knowledge): *methods*, *messages* and *generic methods*. None of them is universal. They all is just a convenience for programmers and not always correspond to how real objects act. If you decide to have a class for a cup – it will have no methods, because the cup can't do anything on it's own. Also, it won't react to any messages. Yet the cup can be a participant of many activities: you can fill it with coffee, drink from it, throw it into your stupid boss etc. So anything you can do with the cup should be described via generic methods. Now if you consider yourself as a human – you can respond to messages, but it's not possible to define a finite set of instance methods on *Human-Being* class, because everything you're capable of and how you behave was learned during your whole life (like realizing that making your chief angry can get you fired). Besides, as for exceptions again, – objects of the same class can act differently (just like people do).

The problem

Semantic networks and frames were the most promising and intensively developed models for knowledge representation. Despite their conceptual simplicity, these formalisms have serious weaknesses and can't cover all possible use cases. So, clearly, there is a need for a new form of knowledge representation.

Let's summarize all apparent requirements, which that model must satisfy:

1) Support for full declarative and procedural knowledge representation.

2) It should be possible to express all kinds of relations, including N-ary (multiple).

3) Support for fuzzy logic (there must be a way to assign "certainty" level to attributes and relations).

4) Multiple inheritance.

5) Prototypes and incomplete types (objects can be constructed even if full class specification isn't known yet).

6) It should be possible to specify value ranges and restrictions for slots.

7) The notation must be serializable and, if possible, have a graphical form.

Semantic frames

When it comes down to practical realization there is almost no difference between semantic nets and frames, because both are usually implemented using the same technique – via hash tables or associative arrays. From this perspective relations and slots are just annotated references (pointers) to other nodes (entities or frames). So, following this logic we can conclude that it's possible to merge both formalisms into one – *semantic frames*, combining their advantages and overcoming limitations by using needed extensions.

There's no need to design the semantic frame model from ground up. In fact, it would be much more easier and cleverer to adapt some already existing technology. The most obvious way is to utilize the power of S-expressions and use one of Lisp dialects to implement a classifier and an inference engine.

From this perspective there can be several approaches to do frames in Lisp programming language:

1) Build frames entirely on top of CLOS (Common Lisp Object System) using macros if necessary.

2) Build frames using associative arrays:

frame = '(', frame name, frame type, slots, ')';
frame name = identifier;
frame type = identifier | **reference**;
identifier = **symbol** | **string** | **keyword**;
slots = '(', { slot }, ');
slot = '(', { pair }, ')';
pair = '(', attribute name, '.', attribute data, ')';
attribute name = 'name' | 'range' | 'default-value' | 'allowed-types' | … .

3) Combined solution: use associative lists for basic frame structure and specialized objects for holding values (references) and specifying their attributes:

frame = '(', frame name, frame type, slots, ')';
frame name = identifier;
frame type = identifier | **reference**;

identifier = **symbol** | **string** | **keyword**;
slots = '(', { slot }, ');
slot = '(', slot name, '.', slot data, ')';
slot name = identifier;
slot data = *Value class instance | Range class instance | … .*

As for graphical notation – it is best to use class diagrams from UML, extending them with additional sections for events and messages.

Conclusion

The progress in AI department heavily relies on achievements from cognitive psychology, which studies mental processes such as attention, language use, memory, perception, problem solving, creativity, and thinking, as well as other fields of science (like quantum physics) and philosophy.

Knowledge is the fundamental and the most important component in AI systems; it has specific characteristics that clearly distinguish it from pure information. There are many knowledge representation models, each has its own advantages and disadvantages, so none of them fully satisfies a well-established set of requirements of AI community.

REFERENCES

1. Башмаков, А.И. Интеллектуальные информационные технологии / А.И. Башмаков, И.А. Башмаков. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2005. – 304 с.
2. Макаренко, С.И. Интеллектуальные информационные системы / С.И. Макаренко. – Стврополь : СФ ММГУ им. М.А. Шолохова, 2009. – 207 с.
3. Information [Electronic resource] / Wikipedia – The Free Encyclopedia. – Mode of access: https://en.wikipedia.org/wiki/Information. – Date of access: 20.11.2016.
4. Perception [Electronic resource] / Wikipedia – The Free Encyclopedia. – Mode of access: https://en.wikipedia.org/wiki/Perception. – Date of access: 21.11.2016.
5. Knowledge [Electronic resource] / Wikipedia – The Free Encyclopedia. – Mode of access: https://en.wikipedia.org/wiki/Knowledge. – Date of access: 24.11.2016.
6. Knowledge representation and reasoning [Electronic resource] / Wikipedia – The Free Encyclopedia. – Mode of access: https://en.wikipedia.org/wiki/Knowledge_representation_and_reasoning. – Date of access: 25.11.2016.
7. Semantic network [Electronic resource] / Wikipedia – The Free Encyclopedia. – Mode of access: https://en.wikipedia.org/wiki/Semantic_network. – Date of access: 27.11.2016.
8. Frame (artificial intelligence) [Electronic resource] / Wikipedia – The Free Encyclopedia. – Mode of access: https://en.wikipedia.org/wiki/Frame_(artificial_intelligence). – Date of access: 28.11.2016.