Semantic networks: visualizations of knowledge

Roger T. Hartley and John A. Barnden

The history of the development of semantic networks is almost as long as that of their parent discipline, artificial intelligence. They have formed the basis of many fascinating, yet controversial, discussions in conferences and in the literature, ranging from metaphysics through to complexity theory in computer science. Many excellent surveys of the field have been written, and yet it is our belief that none of them has examined the important link between their use as a formal scheme for knowledge representation and their more heuristic use as an informal tool for thinking. In our consideration of semantic networks as computerized tools, we will discuss three levels of abstraction that we believe can help us understand how semantic networks are used.

T
he history of the development of semantic networks is well known (for an introduction to semantic networks, see Box 1). Both Sowa\(^1\) and Lehmann\(^2\) have expounded in excellent scholarly fashion as to their origins in the study of language. Their later development as a tool for representing knowledge is also well known\(^3-4\), as is their role in building computerized inference systems\(^5-9\). Indeed, the triad of intelligent thought, logic and language will never be far from our discussion. From all these sources we learn that semantic networks have three main attributes:

1. They originate in the conceptual analysis of language.
2. They can have an expressiveness equivalent to first-order logic, at least (although many do not).
3. They can support inference through an interpreter that manipulates internal representations.

Many people would go further and say that semantic networks are indistinguishable from formal logic representations\(^11\). However, there is something missing here. The visual aspect of the semantic network idea is clearly important. As Sowa says: ‘network notations are easy for people to read’\(^1\) and this pragmatic aspect of the formalism cannot be ignored. According to Sowa: ‘graphs...can keep all the information about an entity at a single node and show related information by arcs connected directly to that node’. In contrast, in symbolic logic notations: ‘the scattering of information not only destroys the readability of the formula, but also obscures the semantic structure of the sentence from which the formula was derived’. So the battle is joined! The visual aspects of the semantic network notation are preferred (at least by Sowa) over the arcane, but more traditional notation of symbolic logic. Interestingly, this traditional notation was invented by C.S. Peirce before he abandoned it in favor of a diagrammatic form\(^1\).

In this paper, we hope to show that this argument is only one component of a larger, more complex one involving the nature of semantics; we will also show how different notations can lead to different systems with different pragmatic uses.

Meaning

The design and use of a knowledge representation revolves around the business of meaning. Actually, one spin-off from studies in natural language provides a good start, namely, the meaning triangle of Ogden and Richards\(^13\). The triangle relates objects in the real world, concepts that correspond to
Box 1. What is a semantic network?

Although this question ought to be easy, in fact, there is much disagreement in the literature. The common elements are clear, however. A semantic network has three characteristics:

1. A way of thinking about knowledge in which there are concepts linked by relationships. Concepts can be concrete, such as objects, or abstract, such as states or actions. 'REX' is a concrete object, 'DOG' is abstract. Relationships are usually binary. 'HAS' relates 'DOG' and 'FUR'. A semantic network is usually thought of as declarative, i.e. it can capture the meaning of a set of true facts. It is true that 'Rex is a dog'.

2. A diagrammatic representation comprising some combination of boxes, arrows and labels. Typically boxes representing concepts are labeled with a word corresponding to the concept, usually a noun for objects and a verb for actions. The arrows are labeled with the relationship between the two concepts at either end of the arrow. The direction of the arrow shows the directionality, if any, of the relation, i.e. 'dog has fur' not 'fur has dog'. They may contain taxonomic links to facilitate reasoning using inheritance and default properties, or exceptions. For example, Rex 'IS-A' (instance of the class of) dog; dogs have fur, therefore, Rex has fur.

3. A computer representation that allows database-like activity and a variety of inferencing techniques using algorithms that operate on the representations. Many systems are implemented using Lisp or Prolog, but there is no inherent advantage in using any language as the directed graph, which is the closest abstract data type to the ideas in a semantic network, is not supported by any common language as a built-in type. Most people agree that a semantic network is meaningless without this algorithmic interpretation.

Some people say that the diagrams themselves are the semantic network, but we think it is more accurate to say that the diagrams represent the semantic network, which is really a network of concepts as held by a cognitive agent. The fact that we can draw them as diagrams, and represent them in computers, makes them extremely useful tools for cognitive psychology and much of artificial intelligence. As we shall see, they also have uses in more informal settings, when they are often called semantic maps.

Secondly, it is symmetrical, giving a central role to the 'concept' as intermediary between the 'symbol' and the 'object'. Thirdly, the dashed line indicates a weaker link than the solid lines. It displays very neatly the diagrammatic features that are unavailable in classic symbolic forms. However, it is also clear that these features can be abused, thus conveying the wrong information. Consider an alternative diagram that is topologically equivalent but sends a distorted message. As a consequence, Fig. 1B is more difficult to read and understand. That diagrams can have a syntax is self-evident; that this syntax has a clear-cut semantic counterpart is less evident. This, we believe, hints at something that most accounts of semantic networks ignore. If, as most people believe, a semantic network is a better way to represent knowledge to a reader (whether human or machine) then where, in fact, does the improvement lie, and are there any limits to this improvement? Furthermore, are there clear-cut advantages when it comes to machine representations, or is it just 'all the same' when knowledge gets digested by the machine?

Levels of representation

Brachman examined five levels of representation that semantic network designers commonly used, and assumed...
Box 2. Meaning

The meaning triangle shows that meaning is a composition of two relations, from the symbols of language to the real world, through a concept-forming agent. However, this is not the only use of the term meaning. In formal semantics, symbolic structures are given meaning through interpretative mappings, or models, where the target of the mapping is some formal construct that uses mathematical (or at least abstract) ideas. The formal construct is often meant to capture the nature of the real world in some way that it is not, in fact, the real world. Typically, this is done for symbolic systems such as formal logic\(^{4}\) and for programming languages\(^{16}\).

**Brachman's levels of representation**

- Linguistic: arbitrary concepts, words and expressions.
- Conceptual: semantics or conceptual relations (cases), primitive objects and actions.
- Epistemological (structural): concept types, conceptual subpieces, inheritance and structuring relations.
- Logical: prepositions, predicates and logical operators.
- Implementational: atoms and pointers.

References


that they were all equivalent, in the sense of having the same ultimate meaning (Box 2).

Clearly, the intention was to show that representations can be 'high-level' or 'low-level', just like computer languages. In this sense, then, logic is the assembler of representation systems, being only one step above the machine level, whereas a natural language is the furthest away from the machine. Thus, a semantic network represented at the conceptual level can be translated into the epistemological level, then to the logical level, and finally to machine data structures.

Whereas we could agree with the intent, it is difficult to see how it fits with the reality of how knowledge representation is typically carried out and, more importantly for us, with the meaning triangle. We think that it is more realistic to make a simple split, as Sowa does, between external representations and internal representations, because there are important differences between human-readable forms and forms that can be stored and processed in a computer's memory.

The meaning triangle does not distinguish among levels of symbol; in some sense, all symbolic representations, whether using linear text, such as logic, or two-dimensional forms such as semantic networks are equivalent. What is more, both logic and semantic networks (at least Sowa's conceptual graphs) have equivalent forms in the other mode. Conceptual graphs have a linear, textual form, and logic has a diagrammatic form – Peirce's existential graphs. (Although Sowa himself sees his conceptual graphs as distinct from other semantic networks, they are clearly in the same family of representation systems, having all the features described in Box 1.) Figure 2A–D shows Sowa's example from Ref. 1 in all four forms. The sentence (abstract graph) in linear form is: 'If a farmer owns a donkey, then he beats it'. Other notations can be seen in Refs 15 and 16.

If we attempt to reconcile these forms in the context of the meaning triangle, we get Fig. 3A for the two systems of conceptual graphs (CGs) and first-order logic (FOL), each with their two symbolic forms.

Thus the symbol systems all have the same meaning, although the method of mapping through the concept node is different in each case, hinting at a similar split in the concept node.

The representations within the circle in Fig. 3A are alternate external forms. We can add the distinction between external and internal representations to this picture. A computer program is a symbol system, just like FOL or CG notation, but before adding these internal representations to the meaning triangle, we believe it is instructive to add a level between machine level and all the external symbolic forms. A programming language is used as an intermediary between concept and machine. The object-oriented languages support abstractions corresponding to Brachman's epistemological level, i.e. types, inheritance, containment, etc. (see Ref. 17 for an excellent exposition of these ideas). What one does in designing a program is to use these abstractions to support perceived relations and structures in the real world. However, programming these relations and structures directly is too difficult, so we need to use a programming language to implement abstract data types (ADTs). The better languages (object-oriented ones) support our ADTs directly and the compiler can then translate these into machine form. It is natural to use a pointer/node abstraction to support a diagrammatic semantic network, but is this the most appropriate form for linear expressions, or is there even a better way to form these abstractions, as in Lendaris's hash-tables (see Box 3)? Clearly, we have choices, governed by considerations of efficiency, size, speed, etc. (the usual computer science concerns). Thus, the internal representations must be mediated, though necessarily, by an ADT, one which is supported by our chosen language. Figure 3B shows these links.
How do computers change things?

We have discussed the meaning triangle which, as we have seen, really relates an ontology of the world with a conceptual schema and an (external) symbol system. However, through the field of artificial intelligence, the computer scientists have added a fourth leg – the internal representation. The advantages of using these internal representations are:

1) They can make the linguistic representations computable. This means that the rules of a formal system can be applied to the representations to explore complex possibilities of inference and even proof. Without the computer, logic would be too hard for describing complex systems. In fact, Prolog, which uses a restricted but useful form of first order logic, can make logic accessible even to people unversed in the arena of the formal calculus.

2) They make persistent databases of knowledge possible. We can store, in a permanent fashion, efficient representations of human memory that permit complex queries and deterministic processing. Again, without them, these applications would not be possible.

3) They can make diagrammatic representations useful over and above their use in small motivational examples. Human-computer interface technology helps here. We believe that this particular aspect of visualization has been neglected when it comes to discussing semantic networks.

Fig. 3 Extensions to the meaning triangle. (A) The meaning triangle can incorporate different forms of symbolic representation, (B) addition of the Abstract Data Type (ADT) level.
Box 3. Implementing semantic networks

Brachman’s lowest level has the machine concepts of ‘atoms and pointers’, betraying his preference for implementations built using Lisp. Any symbol system can be implemented using any data structure with adequate features. Traditional implementations of systems based on logic have been in Lisp because Lisp’s list structures are perfect for representing linear expressions. What is more, pointers (not visible in the source code but present in the programming model) can provide co-references, just like the arrows in a diagrammatic form. Prolog’s internals contain similar hidden pointers. However, other structures can be used. Lendaris uses a hash-table form to represent graphs* and object-based implementations are also available. There is nothing special about atoms and pointers other than ease of implementation. Semantic networks can also be implemented as neural networks and there are interesting similarities and differences between the two.

References


t Trends in Cognitive Sciences Vol. 1, No. 5, August 1997
The 'John' constant symbol (see Ref. 18 for a useful introduction to indexing in logic implementations). Therefore, in implementing logic, we consciously impose a parallel to the computational locality that is assumed to hold for semantic networks.

A full comparison of logic and semantic networks would need to address the way in which semantic networks facilitate inheritance-based reasoning and allow for exceptions to inherited information. Again, there is no real advantage over logic, because logics can be implemented or designed in such a way to have those advantages. One specific recent movement is to use description logics, which are designed specifically to capture some advantages of semantic networks.

**Visualization in concept maps**

Often, diagrams have been advocated as useful brainstorming tools (for example, Ref. 19). Conceptual mapping is a recognized way of getting ideas down on paper so that their relationships can be seen rather than contemplated abstractly. In this sense, the properties of a graphical network become important. Placement of the nodes on the page is important; the more centrally placed a node is, the more 'central' the associated concept is in the cognitive structure. More importantly, the length of the links signifies similarity in concept. This property has been exploited in Pathfinder networks, where a clever algorithm determines the best placement of nodes on the basis of their cognitive 'distance', as reported by a human agent. As mentioned above, this aspect of the diagrammatic form is absent from many descriptions of semantic networks, as if it was somehow irrelevant. At least Sowa attempts to say why the diagram is better than the corresponding FOL form, but then proceeds to give a linear form in his conceptual graph notation that is clearly as good as the diagram! The two are virtually interchangeable (Fig. 5A and B). Perhaps the example is a poor one, since there must be advantages to the diagrammatic form, it is just that no one can quite put their finger on it. How nearness and connectedness figure into the situation is never spelled out.

A concept map is a visualization technique used for brainstorming, for exploration of ideas and for discussing complex systems among the members of a group. Excellent examples of different types of concept maps, and their uses can be found within Ref. 21. Figure 6 shows a 'systems' concept map from Ref. 21, which is meant to be similar to a flowchart. However, there is no discussion about why the diagrams look like they do, or exactly how they convey the required information. A concept map for water can be found within Ref. 22 that looks suspiciously like a semantic network. Indeed, the only real difference between the two, as presented there, seems to be a pragmatic one, and perhaps more formality in the semantic network's choice of relations. Concept maps are used for knowledge acquisition and analysis, and have a useful heuristic nature, whereas, in
general, semantic networks are used for representation in systems, especially computer systems.

It is clear from all the examples of concept maps that their usefulness degrades as the size increases. As Fig. 4 shows, these diagrammatic forms become useless as complexity gets too high, and the linear form can serve just as well (or as badly, since the linear form is always difficult to read). A graph such as that shown in Fig. 7 demonstrates the relative usefulness of the alternate external representations as complexity increases. Diagrams are better than the linear form for reasonably sized chunks of knowledge, but ultimately they both fail because of the limitations of human cognition.

Conclusions
Semantic networks possess several advantages as a tool for knowledge representation. They are easier to read than linear forms (although this advantage is nullified by too much complexity). They also lend themselves to more natural implementations than algebraic or expression-based systems (although these techniques are not confined to semantic networks, and may not be the most efficient methods). Furthermore, they suggest special forms of inference based on link-following (although these forms of inference can also be implemented in other systems).

We have not mentioned an important aspect of semantic networks that deserves a longer discussion than is possible here. The representation of procedural knowledge (knowing 'how', instead of knowing 'that') has been approached from several different angles but has not yet yielded, we believe, to a definitive analysis. Bringing together the ideas of procedural attachment (linking concept boxes to arbitrary procedures to support inference), the inference engine that interprets a network representation and formal semantics of programming languages might be a starting point.

Semantic networks have led a dual existence, as a motivational diagrammatic form of knowledge representation and as an internal, computerized form, suitable for a variety of computational methods. However, although their visual aspect has been acknowledged, it has not been studied extensively enough. Their resurrection as tools for visualization of the structure of knowledge should be explored, and they have much to offer the field of concept mapping in terms of their formal underpinning and the experience of the artificial intelligence community that uses them.

Outstanding questions

- How can semantic networks best represent procedural knowledge?
- Do semantic networks have any measurable benefits for knowledge acquisition over text-based methods?
- Does the visual aspect of semantic networks help or hinder knowledge representation?
- What is the best way to implement a knowledge representation system based on semantic networks?
- Can (and should) the ties to formal logic and logical inferencing be broken to make semantic networks truly distinctive?
- At what level of complexity do the visual aspects of semantic networks break down?

References
21. AI/M Lab.http://classes.uiuc.edu/AIM/Discovery/Mind/c-m2.htm