Pergamon

Information Processing & Management, Vol. 31, No. 2, pp. 233-247, 1995
Copyright © 1995 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0306-4573/95 $9.50 + .00

0306-4573(94)00042-5

A FRAMEWORK FOR THE COMPARATIVE ANALYSIS AND EVALUATION OF KNOWLEDGE REPRESENTATION SCHEMES

R. BINGI,[1] DEEPAK KHAZANCHI,[2] and SURYA B. YADAV[1]
[1]Area of Information Systems & Quantitative Sciences, College of Business Administration, Texas Tech University, Lubbock, TX 79409, U.S.A.
[2]Information Systems Dept., College of Business, Northern Kentucky University, Highland Heights, KY 41099, U.S.A.

(*Received* 7 *December* 1993; *accepted in final form* 13 *June* 1994)

Abstract—This article sets forth a framework for comparing and evaluating knowledge representation schemes based on the requirements for "good" representation discussed in extant literature. The dimensions of the framework suggest that knowledge representation schemes should possess a suitable mix of four basic considerations: Representational adequacy (Variety of Expressiveness, Modularity, Semantics, and Organization of Knowledge); Inference Methods (Reasoning Strategies, Data, Control and Search Strategies); and Inference Requirements (Computational Efficiency, Transparency of line of control, Completeness, and Consistency). A comparative analysis and evaluation of four popular knowledge representation schemes—Logic, Production Rules, Semantic Nets, Frames—highlighting their strengths and weaknesses in the context of the framework is furnished as evidence of its inherent validity and usefulness. In conclusion, it is submitted that incorporating an appropriate blend of the various dimensions elucidated in this article could be a step towards developing more flexible, dynamic, and valuable knowledge representation schemes.

Keywords: Knowledge representation schemes, Framework, Representational properties, Representational adequacy, Logic, Semantic net, Production rules, Frames.

INTRODUCTION

The notion that knowledge representation schemes play an important role in building intelligent systems is widely accepted by both researchers and practitioners of Artificial Intelligence (AI). There is agreement on the fact that intelligence depends on "how" knowledge is stored and handled in the human mind, and hence, in any kind of intelligent device.

Research in the knowledge representation area based on this perspective of AI began only some 30 years ago (Bibel *et al.*, 1990). Knowledge representation has a long tradition rooted mainly in Philosophy, Logic, and Psychology. In fact, both the notions of *knowledge* and *representation* touch on deep philosophical issues. Since AI research methodology involves the design of programs that exhibit intelligent behavior, researchers have often taken a rather pragmatic approach to the subject of knowledge, focusing on improving the behavior of their programs. Schemes and theories of knowledge representation have undergone rapid changes and development in the past 15 years. The concepts of "knowledge," "knowledge representation schemes," "knowledge base," and "knowledge domain" are central to the development of AI systems that organize and apply domain-specific knowledge. In order to provide a common basis for the discussion in this paper, these concepts are defined below.

Knowledge

Knowledge is an integrated collection of facts and relationships which, when exercised, produces competent performance (Rosenberg, 1986). It is having a familiarity with language, concepts, procedures, rules, ideas, abstractions, places, customs, and associations,

Correspondence should be addressed to the second author.

coupled with an ability to use these notions effectively in modeling different aspects of the world (Patterson, 1990).

Knowledge representation schemes

Knowledge representation schemes are a means of encoding and storing knowledge in a database or "knowledge base" (Rosenberg, 1986). These representations could be in many different forms, depending on the type of problem being solved.

Knowledge base

It is the component of an AI system that consists of data structures and procedures that represents domain-specific knowledge for a particular AI application (Barr & Feigenbaum, 1981).

Knowledge domain

It is that problem domain in which a particular chunk of knowledge facilitates effective and efficient problem solving (Chandrasekaran, 1984; Goldstein & Papert, 1977).

In the last 30 years, AI researchers and practitioners have developed several knowledge representation schemes. But, there has been no *a priori* development of a comprehensive set of dimensions to compare (and evaluate) them. In the past, comparison of knowledge representation schemes was either descriptive in nature, listing the advantages and limitations of each without utilizing an organized set of criteria to make the comparison, or only a few issues were considered for the comparison (Baldwin & Kapser, 1986; Barr & Feigenbaum, 1981; Nilsson, 1980). Thus, the purpose of this article is to set forth a *comprehensive framework for the evaluation and comparison of knowledge representation schemes*, and demonstrate its applicability by comparing/evaluating four popular schemes. The framework is logically derived from the characteristics ascribed to "good" knowledge representation schemes by several authors in extant literature. Not only does such a framework help in the organization of the current knowledge regarding requirements for representation schemes, but also serves as a guidepost for future development of better knowledge representation schemes. Furthermore, existing schemes have only been adequate for certain problems or classes of problems, in terms of the range of functionality provided.

Knowledge representation schemes can be categorized into two types: *knowledge structuring schemes* and *implementational schemes*. *Knowledge structuring schemes* such as MOPS (Memory Organization Packet Scheme) and TOPS (Thematic Organization Packet Scheme) are used for organizing knowledge that have some common overall theme or goal-related point (Schank, 1982); *implementational schemes* such as Logic, Production Rules, Frames, and Semantic Nets are used for representing domain knowledge within a computer system. The discussion in this paper is limited to the implementational knowledge representation schemes.

Brachman and Levesque (1986) state that "it is easy to forget that a representational scheme is first and foremost representational, and that consideration of how to tailor a scheme either to machines (at the symbol level) or to people (at the systems engineering level) must be preceded by a good understanding at the knowledge level of what knowledge a scheme can or cannot represent" (p. 12). In keeping with this view, our definition and corresponding analysis of knowledge representation schemes is at the knowledge-level (Newell, 1982). In other words, this paper is not concerned with 'how' a knowledge representation scheme is implemented, but concentrates on the 'kinds of knowledge' that the scheme can represent.

UNDERPINNINGS OF THE FRAMEWORK

The theoretical underpinnings for the framework are the requirements of good knowledge representation. In developing the dimensions of the framework, these requirements have been adapted from the work of many different authors including: Bench-Capon

(1990), Reichgelt (1991), Nisenfeld (1989), Luger and Stubblefield (1989), Rich and Knight (1991), and Winston (1984). Their insight into the essential characteristics of representation schemes provides the basis for the dimensions of our framework.

Characteristics of "good" knowledge representation

Bench-Capon (1990) asserts that knowledge representation schemes need to satisfy *criteria of adequacy*. These criteria relate to things that the scheme must have if it is to do what is required of it. If the representation fails on one of these aspects, whatever virtues it may possess when considered on other grounds will be of no avail. This is because it will not be possible to represent the knowledge we need. The criteria of adequacy need to meet the following three requirements (Bench-Capon, 1990; McCarthy & Hayes, 1969): *Metaphysical adequacy, Epistemic adequacy*, and *Heuristic adequacy*.

Metaphysical adequacy requires that there be no contradiction between the facts we wish to represent and our representation of them. To use an example from McCarthy and Hayes (1969), we could not produce an adequate representation of physical objects in the world if we tried to represent them as a collection of non-interacting particles. Since objects do in fact interact with one another, there is a fundamental contradiction between the "represented" and the "representation," which renders the represented metaphysically inadequate. Note that this criterion does not determine the representation we must use; representation of the world either as a collection of particles that interact through forces between each pair of particles, or a giant quantum mechanical wave function would be metaphysically adequate, in that there is no such contradiction. At this level the representations are mainly useful for constructing general theories.

The requirement of *epistemic adequacy* is that the representation must provide us with the ability to express the facts we wish to express. The third criterion, *heuristic adequacy*, was somewhat tentatively proposed by McCarthy and Hayes (1969), and was meant to suggest the need for the representation to be itself capable of expressing the reasoning that is gone through in solving a problem. This is perhaps the most difficult to fulfill, and it is unclear whether any of the paradigms discussed in this paper would satisfy it completely. It would be a definite requirement if we wished to build systems with a certain degree of "self-consciousness," capable of reflecting on their own reasoning. Thus, representations that are perfectly acceptable in other respects may not be heuristically adequate; this will, of course, limit what can be done with them, but does not make them entirely unusable.

A second set of criteria relate to the desirable aspects of the representation. *Computational tractability* is another requirement of a good knowledge representation scheme (Bench-Capon, 1990). The three criteria discussed under "criteria of adequacy" would hold good for the representation of knowledge for any purpose whatsoever. We are, however, interested in representing knowledge for a particular purpose, namely, to enable its incorporation within a computer system. The *computational tractability* criterion requires that we be able to manipulate the representation efficiently within a computer system. For example, to take an obvious case, although the English language as used by native speakers scores well on the criteria of adequacy, it would be useless as a computer-oriented knowledge representation scheme. This is because it is not, at least currently, possible to use it in computer systems.

Bench-Capon (1990) adds other knowledge representation considerations, such as *lack of ambiguity, clarity, uniformity, notational convenience, relevance, declarativeness*, under a major group called *expressiveness*.

Lack of ambiguity means that every valid expression in the representation should have only one interpretation. This is essentially the requirement that the semantics of the representation be well defined.

Clarity requires that the representation be amenable to understanding by people, even those who may not be entirely immersed in the particular representation formalism.

Uniformity requires that in any given system, the manner of representing a given item of knowledge should not be an arbitrary choice.

Notational convenience may relate to the knowledge we are trying to represent, in the sense that some kinds of knowledge may fit more naturally some representations rather than others. Also, it may be a matter of personal preference on the part of the person building the system or supplying the knowledge. According to this feature it is important to choose a representation that all concerned like and like using.

Relevance suggests that the assessment of a representation cannot be absolute, but is relative to the task we are trying to perform (note: Bench-Capon is referring to the expressiveness of a representation, i.e., expressiveness vs ease of manipulation – a gain in expressiveness results in complexity and is difficult to manipulate).

A *declarative representation* requires that the meanings of statements be independent of the use made of them. Bench-Capon (1990) believes that the ideal of declarativeness must often be compromised, but it is a property to which the representation should aspire.

Reichgelt (1991) discusses four levels of knowledge representation: *implementational level*, *logical level*, *epistemological level*, and *conceptual level*. For each level, he lists desirable criteria of knowledge representation.

Implementation level. Knowledge representation formalisms are intended to be used to represent information on a computer, and it is therefore essential that it be possible to build a computer program underlying the knowledge representation formalism. This is the main concern at the implementational level. The questions that would be asked at this level are "what data structures are best suited for representing expressions in the formalism inside the machine?" and "would this programming scheme lead to a more efficient/more elegant program?" The main criteria of adequacy at this level are (Reichgelt, 1991):

Space efficient: An implementationally adequate knowledge representation formalism should allow one to store information in a space-efficient way.

Time efficient: It should draw its inferences in a time-efficient way (i.e., it should draw its inferences as quickly as possible). One way in which this could be achieved is by having an efficient indexing mechanism.

Logical level. Notationally, the main concern at the logical level is the logical properties of the knowledge representation formalism. The main criteria of adequacy proposed here are:

Clear semantics: The knowledge representation formalism should clearly specify what the meanings are of the syntactically well formed expressions.

Sound inference rules: According to this criterion, if the information that is explicitly stored in the knowledge base is true, then the implicit information that can be retrieved using the inference rules should be true as well. A knowledge representation formalism that allows one to retrieve false information from true information is not satisfactory.

Epistemological level. The main concerns here are the knowledge structuring primitives needed for a satisfactory knowledge representation formalism and the types of inference strategy that should be made available. At this level, one is concerned with discovering the types of primitives needed for representing particular pieces of knowledge without considering which particular primitives are needed. For example, the types of primitives needed for representing the knowledge used in certain skills, such as the knowledge used by an expert pianist, are likely to be very different from those needed for representing the knowledge used by somebody trying to prove a mathematical theorem. Similarly, the types of inference strategies used in different tasks are likely to be very different. For example, somebody trying to diagnose a faulty electronic circuit is likely to use different inference strategies from somebody who is designing a new circuit. It should be noted that at the epistemological level, one does not make any decisions about which actual primitives and inference strategies are used to represent knowledge about some domain. Rather, at this level, one is exclusively concerned with the types of primitive expressions and inference strategies used. The main criterion of adequacy proposed here by Reichgelt (1991) are: *naturalness*, *modularity*, and *granularity*. These will be elaborated in the next Section.

Conceptual level. The conceptual level is concerned with the actual primitives that should be included in a knowledge representation formalism. Suppose, for example, we had chosen, at the epistemological level, that knowledge was to be represented using a semantic network. That is, the decision was made that the primitive in the formalism would be nodes for representing objects, and arcs between those nodes for representing relationships between objects. The decisions at the conceptual level would then concern, for example, which actual arcs there would be. Would there be an *is-a* arc, or *part-of* arc, and so on? The main criterion of adequacy proposed for this level by Reichgelt (1991) is *Conciseness.* Conceptual criteria of adequacy refer to the question "how concisely can particular pieces of knowledge be represented?" If it is impossible to represent a simple piece of knowledge in a concise way, then it is likely that the actual primitives used are wrong. Similarly, if a particular simple inference can only be made in a very complicated way, then one suspects that the actual inference procedure used is not adequate.

Nisenfeld (1989) asserts that a good knowledge representation scheme must satisfy the following five requirements:

1. represent at the proper level all types of application relevant knowledge;
2. manipulate the representation (reason) to yield useful information;
3. explicitly incorporate information that allows the efficient application of knowledge of the problem (i.e., search strategies, heuristic etc.);
4. structure the knowledge in such a way as to allow the system to focus only on the knowledge relevant to a particular situation; and
5. support extensibility, the ability to add new information and delete old information without invalidating the rest of the knowledge base.

Luger and Stubblefield (1989) state that a knowledge representation scheme should have the following characteristics:

1. be adequate to express all of the necessary information;
2. support efficient execution of the resulting code; and
3. provide a natural scheme for expressing the required knowledge.

Rich (1983) suggests that a good system for the representation of knowledge in a particular domain should possess the following four properties:

Representation adequacy. The ability to represent all of the kinds of knowledge needed in that domain;

Inferential adequacy. The ability to manipulate the representational structures in such a way as to derive new structures corresponding to new knowledge inferred from old;

Inferential efficiency. The ability to incorporate into the knowledge structure additional information that can be used to focus the attention of the inference mechanisms in the most promising direction; and

Acquisitional efficiency. The ability to acquire new information easily. The simplest case involves direct insertion, by a person, of new knowledge into the database. Ideally, the program itself would be able to control knowledge acquisition.

Finally, Winston (1984) lists the following desiderata for "good" knowledge representation:

1. good representations make the important things explicit;
2. they expose natural constraints, facilitating some class of computations;
3. they are complete—we can say all that needs to be said;
4. they are concise—we can say things efficiently;
5. they are transparent to us—we can understand what has been said;
6. they facilitate computation—we can store and retrieve information rapidly;
7. they suppress detail—we can keep rarely used information out of sight, but we can still get to it when necessary; and
8. they are computable by an existing procedure.

DIMENSIONS OF THE FRAMEWORK

Based on the previous review of current thought on the characteristics of "good" knowledge representation schemes and our own experience, we recommend the following dimensions for the proposed "framework for comparing and evaluating knowledge representation schemes."

1. *Representational adequacy*

The ability to represent all kinds of knowledge needed in a particular domain. This dimension of the framework is reflected in terms of two attributes: *types of knowledge* that can be represented with the specific knowledge representation scheme, and *granularity*.

1.1 *Types of knowledge.* The AI literature discusses two types of knowledge: *declarative knowledge* and *procedural knowledge. Declarative knowledge* is information about the facts, concepts, and relations of a particular problem domain (Harmon & King, 1985; McCarthy & Hayes, 1969; Winograd, 1975). *Procedural knowledge* is the information about how to reason with the facts, concepts, and relations (Harmon & King, 1985; McCarthy & Hayes, 1969; Winograd, 1975). Clearly both types of knowledge are required to solve problems. However, the kind of declarative and procedural knowledge a system can represent has a direct impact upon the types of problems the system can solve.

The advantages claimed for declarative representation schemes are: the flexibility and economy of the representations, the completeness and certainty of the deductions, and the modifiability of the system. The advantages claimed for procedural representation schemes are the directness of the line of inference (using domain-specific heuristics to avoid irrelevant or unnatural lines of reasoning) and the ease of coding and understandability of the reasoning process itself (Barr & Feigenbaum, 1981). Good representation schemes would allow for some combination of both procedural and declarative knowledge.

1.2 *Granularity or grain size.* Knowledge has to be stored in chunks. The granularity of the knowledge representation determines the size of the chunks in which knowledge is organized (Reichgelt, 1991). One may, for example, store knowledge as a set of facts, thus opting for a knowledge representation with a relatively *fine* granularity; alternatively, one may want to organize in larger chunks (*coarse* granularity) corresponding to what one might call a concept—all the knowledge there is about a particular object, or class of objects. The degree of detail needed depends largely on the performance desired (McCarthy & Hayes, 1969). In general, uniformity of detail across the objects and events seems desirable for a given reasoning task (Bobrow, 1975). Good representation schemes would allow the flexibility of storing knowledge in various degrees of granularity, depending upon the particular problem domain.

2. *Representational properties*

The important representational properties of a formalism are Naturalness, Expressiveness, Modularity, Semantics, and Organization of knowledge. For any formalism, these properties could vary among *excellent*, *average*, or *poor*.

2.1 *Naturalness.* The representation needs to be natural to an expert's thinking. Luger and Stubblefield (1989) state that knowledge representation languages are also tools for helping humans solve problems. As such, a representation should provide a "natural" framework for expressing problem-solving knowledge; it should make that knowledge available to the computer and assist the programmer in its organization. Wolfgram *et al.* (1987), and Reichgelt (1991) assert that naturalness results in easier encoding of the knowledge, and reduces debugging and testing efforts.

2.2 *Expressiveness.* In order to discern the key attributes that make knowledge representation schemes expressive, it is useful to look at the properties of logic. This is because logic is considered a highly expressive knowledge representation scheme. The expressiveness concept when applied to logic has two aspects (Reichgelt, 1991): (a) the ability of logic to express incomplete knowledge, and (b) the fact that there are many different types of "logic" to choose from. The first part of the argument draws attention to the fact that it is very easy to express in logic the information about incompletely known situations (e.g., Moore, 1982). As a result of this, one does not have to represent details that are not (yet) known. Usually, incomplete knowledge refers to disjunctive, negative, and existential knowledge (Reichgelt, 1991). The second part of the argument is that its expressive power is due to the existence of a large number of 'logics', each of which allows the formulation of a special type of knowledge. For example, there are temporal logics that allow one to represent and reason with information about time. In logics of this kind, you can express not only that Harry is bald now, but also that Harry had curly hair in the past, thus increasing the range of information that can be expressed in the scheme (Reichgelt, 1991). Some of the various types of logics mentioned in the literature are: temporal logics (Bench-Capon, 1990; Luger & Stubblefield, 1989; Reichgelt, 1991), deontic logics (Bench-Capon, 1990), Epistemic logics (Reichgelt, 1991), Default and Non-monotonic logics (Reichgelt, 1991), and Multiple-valued logics and Higher-order logics (Luger & Stubblefield, 1989).

In view of the above and the lack of an appropriate definition for expressiveness in extant literature, we suggest factoring the general notion, expressiveness, into four distinct facets. Thus, *expressiveness* consists of: *incomplete expressiveness*, *temporal expressiveness*, *default expressiveness*, and *epistemic expressiveness*.

Incomplete expressiveness: It is the ability of a knowledge representation to represent and reason with incomplete knowledge.

Temporal expressiveness: It is the ability of a knowledge representation to represent and reason with knowledge about time.

Default expressiveness: It is the ability of a knowledge representation to represent and reason with default knowledge.

Epistemic expressiveness: It is the ability of a knowledge representation to represent and reason with knowledge about belief.

Although expressiveness is very desirable, it has a price. Many highly expressive representations are too inefficient for use in certain classes of problems. Thus, sometimes expressiveness must be sacrificed to improve efficiency. This must be done without limiting the representation's ability to capture essential problem-solving knowledge. Optimizing this trade-off is a major task for designers of intelligent systems (Luger & Stubblefield, 1989).

2.3 *Modularity.* Modularity is the ability to add, modify, or delete individual data structures more or less independently of the remainder of the database (Barr & Feigenbaum, 1981). It is likely that the knowledge stored in a knowledge base will change over time, either because the domain of the knowledge changes, or because it turns out that the constructor of the knowledge was mistaken and represented some information that is not true. Thus, if the knowledge representation scheme is modular and if a particular piece of knowledge changes, then only a small part of the knowledge base would have to change (Reichgelt, 1991).

2.4 *Semantics.* Semantics is defined as the specification of how syntactically well formed expressions are to be interpreted (Bench-Capon, 1990; Reichgelt, 1991). One principle often used in defining the semantics of any formal scheme is the Principle of Compositionality or Frege's principle. According to this principle, it should be possible to completely determine the meaning of a complex expression on the basis of the meanings of the simpler expressions that make up the complex expression, and the way in which they have been syntactically combined.

2.5 *Organization of knowledge.* This representational property describes the ability of a knowledge representation scheme to organize the knowledge. For example, frame-based schemes help in organizing knowledge hierarchically and, consequently, have an "excellent" ability to organize the knowledge. It must be noted that good organization need not necessarily imply hierarchy.

3. *Inferential methods supported*

The *diversity* and *number* of inferential methods supported by a knowledge representation scheme adds to its value and flexibility. In general, knowledge representation schemes should support *many types* of inferential methods.

3.1 *Reasoning strategies.* Several reasoning strategies are discussed in the extant literature. They include deductive reasoning (resolution, etc.), inductive reasoning, abductive reasoning, analogical reasoning, intuitive reasoning, forward chaining, backward chaining, bi-directional, probabilistic reasoning, default reasoning, guessing, common-sense reasoning (non-monotonic reasoning), inheritance, and spreading activation.

3.2 *Data and control strategy.* There are two types of data and control strategies: *Separable* and *Mixed*. If the knowledge represented is separate from the techniques or procedures used to manipulate this knowledge, then the data and control strategies are *separable*. If the knowledge represented in the database cannot be distinguished from the manipulation procedures, then the data and control strategies are *mixed*. For procedural representations, the data and control strategy is *generally* mixed. This is because we cannot separate the knowledge from the procedure manipulating it. For example, in a procedural system such as traditional hierarchical programs, the data and control rules are clearly mixed. A separable strategy is possible when the knowledge scheme is modular. Hence, it is possible to achieve a separable data and control strategy for any representation by attaining modularity (Reichgelt, 1991). Thus, naturally procedural schemes such as production systems (PRS) can represent knowledge in IF-THEN procedures that are independent of each other and, in consequence, attain a semblance of modularity. The resulting "modularity" of IF-THEN rules provides the ability to implement computer programs with a considerable degree of separation of knowledge and control knowledge (Luger & Stubblefield, 1989; Wolfgram *et al.*, 1987).

3.3 *Search techniques.* There are many different general-purpose heuristic search techniques available for solving problems. Generate-and-test, hill climbing, breadth-first search, best-first search, problem reduction, constraint satisfaction, and means-end-analysis are all examples of heuristic search strategies or weak methods (Rich, 1983).

4. *Inference requirements*

A knowledge representation scheme's inference system provides the ability to look for a solution by systematically searching through various states of an actual problem (Luger & Stubblefield, 1989). The inference engine applies the knowledge to the solution of actual problems. It is the "interpreter" for the knowledge base. The inference mechanism(s) of a knowledge representation scheme can have *computational efficiency*, a *transparent line of reasoning*, *completeness of representation*, and *consistency* of inferencing. The degree to which each of these *inference requirements* is made obtainable by a scheme can be relatively rated as excellent, average, or poor. It is possible that different inference mechanisms available to a knowledge representation scheme may provide the ability to achieve different degrees of these inference requirements. Thus, in the present article, ratings on this dimension are reflective of an overall assessment of common inference mechanisms used for a given knowledge representation scheme.

4.1 *Computational efficiency.* This is the ability of a scheme to manipulate the representation efficiently within a computer system. There is a trade-off between the expressiveness of a representation and its computational efficiency. The more expressive a representation scheme is, the more computationally intractable are its inference rules (Levesque & Brachman, 1985). By relinquishing some of the expressiveness, a representation can gain an increase in computational efficiency. However, it is difficult to decide how much expressiveness we need to relinquish in order to gain an acceptable efficiency.

4.2 *Transparency of line of reasoning.* It is the ability to follow the flow of control in problem solving (Barr & Feigenbaum, 1981). Transparency inherently implies the ability of a user to view and comprehend the stream of reasoning – leading to greater user confidence in the generated solution. Thus, representation schemes that can provide a transparent line of reasoning afford users a clear understanding of their searches for solution(s).

4.3 *Completeness.* Completeness refers to the "amount of knowledge" needed to solve a problem adequately and the representation selected to support that knowledge. The completeness of the representation is dependent not only upon the nature of the relevant system, but also upon the problem-solving strategy(-ies) selected (Nisenfeld, 1989). For example, in the case of a logic representation, resolution and *modus ponens* are examples of inference rules that are sound and, when used with certain appropriate strategies, complete (Luger & Stubblefield, 1989). Completeness is not necessarily always desirable. There are cases when we might want the system to work quickly and not spend a long time finding a particular answer or concluding that it cannot find the answer (Barr & Feigenbaum, 1981).

4.4 *Consistency of inferencing.* A system (deductive) is consistent if all its deductions are correct—that is, if the conclusions are not contradictory and necessarily follow from the premises (Barr & Feigenbaum, 1981; Mendelson, 1964). This implies an ability to infer new correct conclusions from a set of true premises in such a way that these new expressions are consistent with all previous interpretations of the original set of premises. Tightly controlled reasoning in the presence of inconsistency can assist in avoiding the pitfalls of potentially contradictory consequences (Luger & Stubblefield, 1989). Like completeness, consistency is not necessarily always desirable. It is argued that much of our reasoning is done by revising our beliefs in the presence of new information. Thus, it appears that most of our knowledge is not absolute; we regularly accept caveats and exceptions (Barr & Feigenbaum, 1981).

COMPARISON/EVALUATION OF KNOWLEDGE REPRESENTATION SCHEMES

This section uses the framework dimensions established earlier to analyze four knowledge representation schemes: Logic, Semantic Nets, Production Rules, and Frames. The schemes were selected for comparison (and evaluation) because of their wide usage in expert system design (Wolfgram *et al.*, 1987). Figure 1 summarizes the results of a comparative analysis and evaluation of the schemes. The rationale for the ratings in this figure is elucidated in the ensuing discussion of the characteristics (vis-à-vis the framework dimensions) of each scheme and their advantages and disadvantages.

Logic

Logic is one of the oldest formal mathematical and philosophical representation models of knowledge and thought. It is also one of the most developed problem-solving paradigms with a complete and concise vocabulary and syntax. The form of logic most often used is first-order predicate logic, which is an extension of predicate calculus. Logic schemes represent declarative knowledge in the form of independent well formed formulas (wffs) (see, e.g., Luger & Stubblefield, 1989; Rich, 1983). A proposition itself is made up of two parts, a predicate and a subject. For example, MANAGER(DAVIS) is a proposition consisting of the predicate MANAGER and the subject DAVIS. Procedural knowledge consists of the rules of inference, and is usually represented as some general procedure. Logic-based systems typically solve problems by constructing proofs from an initial set of stated conditions and a goal (Baldwin & Kasper, 1986).

Logic is a very general language which, by its own design, provides a clearly defined interface between the semantic intent of an expression and its symbolic representation. The theory of logic is well founded and has been actively studied, producing many efficient computational schemes that guarantee the consistency and soundness of resulting inferences (Chang & Lee, 1973; Wos *et al.*, 1984).

Search strategies supported are breadth-first strategy, set of support strategy, unit preference strategy, and line input form strategy (Luger & Stubblefield, 1989). A combination of strategies can be quite effective in controlling search—for instance, the use of set of support and unit preference. Search heuristics can also be used. An exhaustive set of strategies or even the most sophisticated schemes can be found in the literature (see, e.g., Wos *et al.*, 1984).

CRITERIA	LOGIC	SEMANTIC NETS	PRODUCTION RULES	FRAMES
1. Representational adequacy				
1.1 Kinds of knowledge	Declarative	Declarative	Procedural	Declarative & Procedural
1.2 Granularity	Fine	Fine	Fine	Coarse
2. Representational properties				
2.1 Naturalness	Excellent	Excellent	Excellent	Excellent
2.2 Expressiveness:				
2.2.1. Incomplete	Excellent	**	Poor	Poor
2.2.2. Temporal	Excellent	**	**	**
2.2.3. Default	Excellent	Excellent	**	Excellent
2.2.4. Epistemic	Excellent	Excellent	**	**
2.3 Modularity	Excellent	Poor	Excellent	Average
2.4 Semantics	Excellent	Poor	Average	Poor
2.5 Organization of Knowledge	Poor	Excellent	Poor	Excellent
3. Inference Methods				
3.1 Reasoning Strategies	Many types	Many types	Many types	Many types
3.2 Data and Control Strategies	Separable	Mixed	Separable	Mixed
3.3 Search Techniques	Many types	Many types	Many types	Many types
4. Inference Requirements				
4.1 Computational Efficiency	Poor	Average	Below average	Excellent
4.2 Transparency of Line of Control	Poor	Average	Poor	Average
4.3 Completeness	Excellent	Poor	Poor	Average
4.4 Consistency	Excellent	Poor	Poor	Average

** Criterion is not relevant to this scheme.

Fig. 1. Comparative analysis and evaluation of knowledge representation schemes.

A logic system has a number of important advantages. The primary strength of logic is its expressive power (Levesque, 1984). Logic is very good in handling incomplete knowledge. Also, various extensions like Temporal logics, Default logics, and Epistemic logics enhance the expressiveness of predicate calculus. A second advantage of logic is that the rules of inference are sound and complete (Rich, 1983). It should be noted, however, that in practice most systems modeled after logic do not utilize the complete set of inferences rules and, thus, do not exhibit the completeness characteristic (Rich, 1983). The third advantage of logic is modularity (Barr & Feigenbaum, 1981: Wolfgram *et al.*, 1987). Statements can be added, deleted, or modified without having to consider the impact on the other statements in the representation—akin to the modularity of production rules. Also, statements can be made without having to worry about the context in which they will be used (Wolfgram *et al.*, 1987). Logic often seems a natural way to express certain notations (Barr & Feigenbaum, 1981). As McCarthy (1977) and Filman (1979) point out, the expression of a problem in logic often corresponds to our intuitive understanding of the domain. Finally, logic systems have clear semantics (Reichgelt, 1991).

One of the major disadvantages of logic stems from the separation of representation and processing. The difficulty with most current AI systems lies in the heuristic part of the system—that is, in determining how to use the facts stored in the system's data structure, not in deciding how to store them (Barr & Feigenbaum, 1981). The second disadvantage of predicate calculus is that as the number of facts in the knowledge base increases, the number of ways to combine the facts to make inferences explodes exponentially. Reasoning with predicate calculus is done by using rules of inference, proof by refutation, or resolution theorem proving. Unfortunately, the resolution method of theorem proving is too indiscriminate and the search process cannot be constrained (Barr & Feigenbaum, 1981; Wolfgram *et al.*, 1987). Furthermore, the search process is infeasible (Fedorowicz & Williams, 1986). There do exist a number of strategies that begin to reduce the complexity of the solution search process by guiding, deleting, or editing the possible resolutions. However, no known strategy provides an entirely satisfactory approach (Fedorowicz & Williams, 1986). The third disadvantage of logic is that the problem-solving process (theorem proving) may become opaque (poor transparency of line of control) and obscure (Wolfgram *et al.*, 1987). The fourth disadvantage of logic is its strict utilization of the rules of inference. These rules permit only a deductive form of reasoning based on the truth value of premises (initial conditions) or previously derived theorems. Default reasoning and guessing, which allows conclusions to be drawn from statements with undecided truth values are not permitted (Rich, 1983; Mylopoulos, 1981). For the same reasons, probabilistic reasoning is not supported in logic systems (Baldwin & Kasper, 1986). Even though first-order predicate calculus cannot express default, temporal, and epistemic knowledge, the extensions to it, namely, default logics, temporal logics, and epistemic logics would extend the expressive power. The fifth disadvantage of logic is that it lacks facilities for organizing knowledge (Tanimoto, 1990). The final disadvantage of logic is its poor computational efficiency because of its high expressiveness. Horn clauses restrict its expressiveness and increase the computational efficiency (Bench-Capon, 1990).

Finally, though the logic knowledge representation scheme has a number of disadvantages, the expressive power of the scheme, the soundness and the completeness of the rules of inference, and its large exposure in a number of disciplines make logic a popular representation scheme.

Semantic nets

Semantic nets (also called semantic networks) are the most general representational structure and serve as the basis for other knowledge representations. Semantic nets alone are never directly used to model knowledge. The lack of formal definitive structural rules makes semantic networks not very elegant to use; however, they need to be understood since they constitute the theoretical underpinnings for other data representational methodologies, such as frames and production rules (Wolfgram *et al.*, 1987).

A semantic net is a collection of objects (commonly referred to as nodes) linked together by a relationship. Nodes not only represent objects, but concepts and situations as well, and they are shown graphically by dots, circles, or boxes. Relationships between objects are expressed as arcs or links, and are represented graphically by an arrow between the two nodes for which the relationship is intended to be expressed. Semantic nets can express declarative knowledge naturally.

In terms of logic, the semantic net and some forms of the frame knowledge representation scheme can be described as unary and binary propositions, utilizing a restricted set of predicates (Levesque, 1984; Nilsson, 1980). Predicates common to semantic nets include: IS_A, IS_SPECIALIZATION_OF, CAUSES, and HAS_PART (Barr & Feigenbaum, 1981; Brachman, 1977; Duda *et al.*, 1978; Mylopoulos, 1981). This results in an average level of expressiveness and increased computational efficiency compared to logic.

Restrictions on the expressiveness of declarative knowledge and the lack of procedural knowledge standards present problems for semantic net schemes. Restricted expressiveness is an obvious disadvantage. The lack of procedural standards means that semantic network reasoning is not altogether sound or complete (Baldwin & Kasper, 1986). In semantic network representations, there is no formal semantics (Barr & Feigenbaum, 1981; Reichgelt, 1991).

The advantage of semantic nets is their ability to inherit relationship from other nodes—more specifically, the ability to reason and make assertions about one node and its relationship with another node where no direct arc exists between the two nodes. A basic strength of semantic net is its natural ability to represent "deep knowledge." All arcs are labeled so that the relationships between facts and/or concepts are explicitly defined. Furthermore, graph search schemes allow questions of the form, 'What is the relationship between A and B?'.

A number of systems, for example, HEARSAY and SRI, have been developed using semantic nets with heuristic searches. For reasoning, Quillian (1968) uses the spreading activation model. Most semantic nets use 'matching' as the reasoning mechanism (Barr & Feigenbaum, 1981). Default reasoning (through inheritance), probabilistic reasoning, as well as logical deductive reasoning are also supported (Barr & Feigenbaum, 1981; Brachman, 1977; Duda *et al.*, 1978; Mylopoulos, 1981). Data and control strategy in semantic nets is mixed (Wolfgram *et al.*, 1987). Researchers have made various proposals to extend the expressive power of semantic nets, for example, partitioned networks (Hendrix, 1975, 1979), SNePS networks (Maida & Shapiro, 1982; Rapaport, 1986; Shapiro, 1979), and KL-ONE (Brachman, 1979). Partitioned networks could be used to express epistemic knowledge.

Production rules

A production rule system (PRS) consists of a rule base, an inference engine, and a database (Barr & Feigenbaum, 1981). The rule base consists of independent rules of the form, IF ⟨situation⟩ THEN ⟨action⟩. In general, these rules represent heuristics for solving a particular class of problems. The inference engine contains procedural knowledge about a strategy for selecting a particular production rule to fire. For a rule to be a candidate for firing, the ⟨situation⟩ component of the rule must occur. The database is then modified according to the specifications of the ⟨action⟩ component. The problem-solving process consists of selecting and firing rules until some part of the database matches the goal the system is attempting to achieve. PRSs are relatively efficient in many domains and are especially adaptable to problems naturally expressed in terms of procedures. In production rule representations, the procedures that implement the control cycle are separated from the production rules themselves, thus resulting in a separable control and data strategy (Luger & Stubblefield, 1989; Wolfgram *et al.*, 1987).

The advantage of PRS is the ability to utilize multiple forms of procedural knowledge. These systems are capable of probabilistic reasoning, and default reasoning and guessing (Davis *et al.*, 1977; Mylopoulos, 1981; Rich, 1983; Stefik *et al.*, 1982). However, like semantic nets, the reasoning is not guaranteed to be sound or complete (Baldwin & Kasper, 1986). Forward reasoning, backward reasoning, and a combination of both are supported (Fedorowicz & Williams, 1986).

A special problem of PRSs is inefficiency in large domains (Barr & Feigenbaum, 1981). Since the knowledge of a PRS is represented in modular format, the search efficiency is critically dependent upon the inference engine's ability to select the correct rules to fire. When the knowledge domain is large, the probability of selecting the correct rule is decreased. Therefore, the inference engine's efficiency may negate the efficiency gained by using problem-solving heuristics (Baldwin & Kasper, 1986).

The primary advantage of a PRS is modularity (Barr & Feigenbaum, 1981). Modularity of the rules is important in building the large rule bases of current AI systems—knowing what a proposed rule will mean, in whatever situation it is used, makes the creation of the database much easier. There are indications, however, that modularity is harder to maintain as one moves to larger systems (Rychener, 1976), and even if modularity can be preserved, strongly constraining interaction between rules leads to inefficiencies that might become important problems in large systems (Wolfgram *et al.*, 1987).

The second advantage of the production-system formalism is the ease with which one can express certain important kinds of knowledge. In particular, statements about what to do in predetermined situations are naturally encoded into production rules. Furthermore, it is these kinds of statements that are most frequently used by human experts to explain how they do their jobs (Barr & Feigenbaum, 1981).

The disadvantage of the production-system formalism is that it is hard to follow the flow of control in problem solving—algorithms are less apparent than they would be if they were expressed in a programming language. In other words, although situation-action knowledge can be expressed naturally in production systems, algorithmic knowledge is not expressed naturally. Two factors that contribute to this problem are the isolation of productions (they do not call each other) and the uniform size of productions (there is nothing like a subroutine hierarchy in which one production can be composed of several productions). Function calls and subroutines, common features of all programming languages, would help to make the flow of control easier to follow (Barr & Feigenbaum, 1981). The ability to express incomplete knowledge by production rules is very poor (difficult to express, if not impossible) (Reichgelt, 1991).

Frames

Frames, as in frames of reference, consist of a collection of slots that contain attributes to describe an object, a class of objects, a situation, an action, or an event. Frames provide a concise, structural representation of useful relations that capture the way an expert typically thinks about the data in the knowledge base (Fikes & Kehler, 1985; Wolfgram *et al.*, 1987). Frames are a natural way of retaining the advantages and reducing the weaknesses of production rules and semantic nets (Chandrasekaran, 1984). The frames can represent both declarative and/or procedural knowledge (Barr & Feigenbaum, 1981; Rich, 1983). Frames help in organizing the knowledge in hierarchical structures, in units called chunks (Reichgelt, 1991). The main inference technique used is inheritance in conjunction with default values, embedded procedures, and the use of heuristics (Patterson, 1990; Reichgelt, 1991).

The first key advantage of this scheme, relative to other knowledge representation schemes, is that a frame may be more efficient in representing and maintaining the knowledge (Barr & Feigenbaum, 1981; Bobrow & Winograd, 1977; Mylopoulos, 1981). Reichgelt (1991) asserts that frame-based representation schemes relinquish some of the expressive power (compared to logic based representations), but gain in efficiency. For many applications, they seem to strike the right balance. Also, frame-based representations allow their knowledge bases to be concise and compact (Wolfgram *et al.*, 1987). Frames allow for layers of abstraction to separate out low-level details from high(er)-level ones (Wolfgram *et al.*, 1987). Similar to semantic nets, links between frames guide the reasoning process. Once a relevant frame has been found, procedures specific to the frame control the reasoning process.

The second major advantage for frame-based representations is that procedural knowledge is not constrained to any predefined type; therefore, probabilistic reasoning, guessing, and default reasoning can be supported (Barr & Feigenbaum, 1981; Bobrow & Winograd, 1977; Mylopoulos, 1981). The data and control strategy used with frames is mixed (Wolfgram *et al.*, 1987). The key disadvantage of frame-based schemes is the absence of clear semantics and the limited ability to express incomplete knowledge (Reichgelt, 1991).

IMPLICATIONS AND CONCLUDING REMARKS

To recapitulate, this paper developed and demonstrated a framework for the comparative analysis and evaluation of knowledge representation schemes. The framework is predicated on the desirable characteristics of knowledge representation schemes discussed in extant practitioner and theoretical research. The framework is demonstrated by comparing and evaluating four popular knowledge representation schemes: Logic, Semantic Nets, Production Rules, and Frames.

The comparative analysis and evaluation of the four popular knowledge representation schemes provides the basis for assisting in (a) evaluating a knowledge representation scheme against the requirements of a "good" knowledge representation scheme; and (b) comparing one knowledge representation scheme to another. By comparing each scheme individually against the framework, indirect comparisons of representation schemes against each other can be easily provided. The many other existing knowledge representation schemes may be similarly compared within the framework and, thus, against all other schemes. Thus, for example, one could compare logic to frames on the "semantics" aspect and discover that logics have clearer semantics compared to frames. This type of comparison among various knowledge representation schemes could aid designers in easily visualizing the differences among candidate knowledge representation schemes, and help in selecting an appropriate scheme to meet their specific problem needs. Similarly, a researcher who has developed a new knowledge representation scheme could evaluate the newly developed scheme against the characteristics of good knowledge representation schemes illustrated in the framework developed in this paper. The evaluation could assist in improving the scheme.

Our analysis indicates that none of the more "popular" knowledge representation schemes have all the desirable characteristics ascribed to good knowledge representation. The ultimate implications are twofold. First, that there is a need for a new, more complete, "unified" knowledge representation scheme based on a sound semantic theory. Such a representation scheme would not only be flexible—applicable freely to classes of problems, but also be dynamic—capable of being used by an expert, either human or machine (Luger & Stubblefield, 1989). And secondly, our framework provides a useful basis for developing and comparing new/old knowledge representation schemes.

In conclusion, it should be noted that the notion of mixed representational languages has been discussed and developed in the literature of the last decade. KRYPTON (Brachman *et al.*, 1983, 1985), Cake (Rich, 1982, 1985), and KL-TWO (Vilain, 1985) are all examples of languages that attempt to incorporate the virtues of the various extant representation schemes into a new hybrid formalism. There are many more advantages claimed for these mixed languages (see, e.g., Vilain, 1985). However, this trend does not obviate the need for a knowledge representation scheme that has the desirable characteristics discussed in our framework. In fact, it is quite conceivable that such an organized set of dimensions may have been useful to developers and researchers involved in the development of hybrid representation schemes.

REFERENCES

Baldwin, D., & Kasper, G.M. (1986). Toward representing management-domain knowledge. *Decision Support Systems, 2*(2), 159–172.

Barr, A., & Feigenbaum, A.E. (1981). *The handbook of artificial intelligence*, Vol. 1. Los Altos, CA: William Kaufman.

Bench-Capon, T.J.M. (1990). *Knowledge representation: An approach to artificial intelligence* (A.P.I.C. series no. 32). San Diego, CA: Academic Press.

Bibel, W., Schneeberger, J., & Elver, E. (1990). Representation of knowledge. In H. Adeli (Ed.), *Knowledge engineering fundamentals*, Vol. 1. NY: McGraw Hill.

Bobrow, D.B. (1975). Dimensions of representation. In D.G. Bobrow & A. Collins (Eds.), *Representation and understanding studies in cognitive science*. New York: Academic Press.

Bobrow, D.G., & Winograd, T. (1977). An overview of KRL, a knowledge representation language. *Cognitive Science, 1*(1), 3–46.

Brachman, R. (1977). What's in a concept: Structural foundations for semantic networks. *International Journal of Man-Machine Studies, 9*(2), 127–152.

Brachman, R. (1979). On the epistemological status of semantic networks. In N. Findler (Ed.), *Associative networks: Representation and use of knowledge by computers*. New York: Academic Press.

Brachman, R., Fikes, R., & Levesque, H. (1983). KRYPTON: A functional approach to knowledge representation. *IEEE Computer, 16*(10), 67–73.

Brachman, R., Gilbert, V., & Levesque, H. (1985). An essential hybrid reasoning system: Knowledge and symbol level accounts of KRYPTON. *IJCAI, 9*, 532–539.

Brachman, R., & Levesque, H. (1986). The knowledge level of a KBMS. In M.L. Brodie & J. Mylopoulos (Eds.), *On knowledge based management systems: Integrating artificial intelligence and database technologies* (pp. 9–12). New York: Springer-Verlag.

Chandrasekaran, B. (1984). Expert systems: Matching techniques to tasks. In W. Reitman (Ed.), *Artificial intelligence applications for business* (pp. 41–85). Norwood, NJ: Ablex.

Chang, C., & Lee, R.C. (1973). *Symbolic logic and mechanical theorem proving*. New York: Academic Press.

Davis, R., Buchanan, B., & Shortlife, E. (1977). Production rules as a representation for a knowledge-based consultation program. *Artificial Intelligence, 8*(1), 15–45.

Duda, R.O., Hart, P.E., Nilsson, N.J., & Sutherland, G.L. (1978). Semantic network representations in rule-based inference systems. In D.A. Waterman and F. Hayes-Roth (Eds.), *Pattern directed inference systems* (pp. 203–221). New York: Academic Press.

Fedorowicz, J., & Williams, G.B. (1986). Representing modeling knowledge in an intelligent decision support system. *Decision Support Systems, 2*(1), 3–14.
Fikes, R., & Kehler, T. (1985). The role of frame-based representation in reasoning. *Communications of the ACM, 28*, 904–920.
Filman, R.E. (1979). The interaction of observation and inference in a formal representation system, *IJCAI, 6*, 269–274.
Goldstein, I., & Papert, S. (1977). Artificial intelligence, language and the study of knowledge. *Cognitive Science, 1*(1), 84–123.
Harmon, P., & King, D. (1985). *Expert systems, artificial intelligence in business.* New York: Wiley.
Hendrix, G. (1975). Extending the utility of semantic networks through partitioning. *IJCAI, 4*, 115–121.
Hendrix, G. (1979). Encoding knowledge in partitioned networks. In N. Findler (Ed.), *Associative networks: Representation and use of knowledge by computers.* New York: Academic Press.
Levesque, H.J. (1984). A fundamental tradeoff in knowledge representation and reasoning. *Fairchild Technical Report*, No. 658.
Levesque, H.J., & Brachman, R. (1985). A fundamental tradeoff in knowledge representation and reasoning. In R. Brachman and H. Levesque (Eds.), *Readings in knowledge representation* (Revised edition), New York: Academic Press.
Luger, G.F., & Stubblefield, W.A. (1989). *Artificial intelligence and the design of expert systems.* San Francisco, CA: Benjamin Cummins.
Maida, A., & Shapiro, S. (1982). Intentional concepts in propositional semantic networks. *Cognitive Science, 6*, 291–330.
McCarthy, J. (1977). Epistemological problem of artificial intelligence. *IJCAI, 5*, 1038–1044.
McCarthy, J., & Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie (Eds.), *Machine intelligence*, Vol. 4 (pp. 463–502). Edinburgh, Australia: Edinburgh University Press.
Mendelson, E. (1964). *Introduction to mathematical logic.* New York: D. Van Nostrand.
Moore, R.C. (1982). The role of logic, knowledge representation, and common sense reasoning. *Proceedings of the AAAI-82.*
Mylopoulos, J. (1981). An overview of knowledge representation. *SIGART Newsletter, 74*, 5–12.
Newell, A. (1982). The knowledge level. *Artificial Intelligence, 18*(1), 87–127.
Nilsson, N.J. (1980). *Principles of artificial intelligence.* Palo Alto, CA: Tioga.
Nisenfeld, A. (1989). *Artificial intelligence handbook*, Research Triangle Park, NC: Instrument Society of America.
Patterson, D.W. (1990). *Introduction to artificial intelligence and expert systems.* Englewood Cliffs, NJ: Prentice Hall.
Quillian, M. (1968). Semantic theory. In M. Minsky (Ed.), *Semantic information processing.* Cambridge, MA: MIT Press.
Rapaport, W. (1986). Logical foundations for belief representation. *Cognitive Science, 10*, 371–422.
Reichgelt, H. (1991). *Knowledge representation: An AI perspective.* Norwood, NJ: Ablex.
Rich, C. (1982). Knowledge representation languages and predicate calculus: How to have your cake and eat it. *AAAI, 82*, 193–196.
Rich, C. (1985). The layered architecture of a system for reasoning about programs. *IJCAI, 9*, 540–546.
Rich, E. (1983). *Artificial intelligence.* New York: McGraw-Hill.
Rich, E. & Knight, K. (1991). *Artificial intelligence.* NY: McGraw-Hill.
Rosenberg, J.M. (1986). Dictionary of artificial intelligence and robotics. New York: Wiley.
Rychener, M.D. (1976). *Production systems as a programming language for artificial intelligence applications.* Pittsburgh, PA: Computer Science Department, Carnegie Mellon University.
Schank, R.C. (1982). Reminding and memory organization: An introduction to MOPs. In W.G. Lehnert & M.H. Ringle (Eds.), *Strategies for natural language processing* (pp. 455–493). Hillsdale, NJ: Lawrence Erlbaum.
Shapiro, S. (1979). The SNePS semantic network processing system. In N. Findler (Ed.), *Associative networks: Representation and use of knowledge by computers.* New York: Academic Press.
Stefik, M., Aikins, J., Balzer, R., Benoit, J., Birnbaum, L., Hayes-Roth, F., & Sacerrdoti, E. (1982). The Organization of expert systems, A tutorial. *Artificial Intelligence, 18*(2), 135–173.
Tanimoto, S.L. (1990). *The elements of artificial intelligence using common LISP.* New York: Computer Science Press.
Vilain, M. (1985). The restricted language architecture of a hybrid representation language. *IJCAI, 9*, 547–551.
Winograd, T. (1975). Frame representation and the declarative/procedural controversy. In D.G. Bobrow & A. Collins (Eds.), *Representation and understanding* (pp. 185–210). New York: Academic Press.
Winston, P.H. (1984). *Artificial Intelligence.* New York: Addison-Wesley.
Wolfgram, D.D., Dear, T.J., & Galbraith, C.S. (1987). *Expert systems for the technical professional.* New York: Wiley.
Wos, L., Overbeek, R., Lusk, E., & Boyle, J. (1987). *Automated reasoning, introduction and applications.* Englewood Cliffs, NJ: Prentice-Hall.