This paper presents a comprehensive survey and analysis of semantic network representation schemes (circa 1979). The survey pinpoints a large number of the seminal contributions to the history of semantic networks. It also helps remove some of the confusion about such schemes by illustrating how modern semantic networks evolved first from strict psychological models, and then from linguistic structures, neither of which methodologies intended to produce representation languages. Brachman's subsequent analysis involves the level of primitives used in the various networks. He proposes that the similarity among network notations is illusory, and that there are really at least five different types of links used. The levels range from implementational, simple pointers in the memory of a computer, all the way to linguistic, links whose meaning is tied to natural language. Brachman also distinguishes between logical primitives and epistemological ones. The latter form the basis for KLONE (subsequently KL-ONE), a representation language that Brachman sketches in the third part of the paper (see [Brachman and Schmolze 85] for a much more extensive introduction to that system and [Brachman et al., Chapter 24] for some recent KL-ONE-based developments). While KL-ONE has received much attention and has been a significant contribution to the field, this paper is worth reading for the survey alone.
ON THE EPISTEMOLOGICAL STATUS OF SEMANTIC NETWORKS*

Ronald J. Brachman

ABSTRACT

This chapter examines in detail the history of a set of network-structured formalisms for knowledge representation—the so-called "semantic networks." Semantic nets were introduced around 1966 as a representation for the concepts underlying English words, and since then have become an increasingly popular type of language for representing concepts of a widely varying sort. While these nets have for the most part retained their basic associative nature, their primitive representational elements have differed significantly from one project to the next. These differences in underlying primitives are symptomatic of deeper philosophical disparities, and I discuss a set of five significantly different "levels" at which networks can be understood. One of these levels, the "epistemological," or "knowledge-structuring," level, has played an important implicit part in all previous notations, and is here made explicit in a way that allows a new type of network formalism to be specified. This new type of formalism accounts precisely for operations like individuation of description, internal concept structure in terms of roles and interrelations between them, and structured inheritance. In the final section, I present a brief sketch of an example of a particular type of formalism ("Structured Inheritance Networks") that was designed expressly to treat concepts as formal representational objects. This language, currently under development, is called KLONE, and it allows the explicit expression of epistemological level relationships as network links.

INTRODUCTION

The idea of a memory based on the notion of associations is apparently a very old one—Anderson and Bower [1973] trace the idea back as far as Aristotle. However, only in the last ten years has the associative memory idea taken a firm hold with those interested in modeling human memory or providing working memories for intelligent computer programs. Yet in the short time since Ross Quillian first introduced the idea of a "semantic network" in his Ph.D. thesis [1966], network models of information and their computer implementations have become rampant.

While Quillian's original intent was to represent the semantics of English words in his nets, representations that looked very similar were soon being used to model all sorts of nonsemantic things (e.g., propositions, physical object structure, "gated one-shot state coupling"). Yet, virtually every networklike formalism that has appeared in the literature since 1966 has at one time or another been branded by someone a semantic net. The possibility of confusion over the real nature of the network slipped by virtually unnoticed, since everyone working in the area already "knew" with what they were working. But as interest has developed over the last two years in the semantics of the semantic net itself, the epistemological status of the representation has become increasingly suspect. This chapter is an attempt to clear up what we mean when we call our representations semantic nets, and to examine what claims about the structure of knowledge these so-called representations actually make.

To this end, I shall first examine the history of semantic networks, covering as broadly as possible in a limited space the major developments in associative memory models from 1966 through the present. I shall then attempt to explain why so many of the earlier formalisms are inadequate in several ways, and why the more recent ones approaching complete logical adequacy are perhaps not as useful for knowledge representation as it was hoped they might be. The substance of this analysis will be a close look at the kinds of entities chosen by designers to be primitive in their network schemes. By elucidating the different kinds of primitives employed in these nets, I hope to illustrate how there are at least five different kinds of knowledge that have become confusingly called "semantic" by their association with semantic networks. For the purposes of the analysis, I shall postulate and discuss five levels of semantic net primitive corresponding to these kinds of knowledge—the "implementational," "logical," "epistemological," "conceptual," and "linguistic" levels.

Of these levels has been less used and understood than the others, but may have significantly more utility in the near future for general knowledge representation tasks than the others. This is the epistemological level of knowledge representation—the one on which several new nonnetwork formalisms like KRL [Bobrow and Winograd, 1977] and ERF [Goldstein and Roberts, 1977; Roberts and Goldstein, 1977] are built, and the one dealing with things like "inheritance," "abstraction," and concept structuring. In Section 3, I examine some of the implications of this level of thinking, and show how it has influenced my own work on what I used to think of as semantic networks. I shall present some of the prominent aspects of a new netlike formalism—the "Structured Inheritance Network." A structured inheritance network (SI-Net) has a fixed set of node and link

* Prepared in part at Bolt Beranek and Newman Inc. under contracts sponsored by the Defense Advanced Research Projects Agency and the Office of Naval Research. The views and conclusions stated are those of the author and should not be interpreted as necessarily representing the official policies, either express or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.
types (the number of which is small), thereby providing the basis for a fixed and well-defined interpreter. The links in this kind of net are used to explicitly express "epistemological" relationships between Concepts and their parts ("Roles" and "Structural Descriptions"), that is, structuring relationships between formal objects used to represent knowledge. SI-Nets can be used to illustrate the level of concern of epistemological formalisms in general, and I shall use a particular SI-Net language, called KLONE, to help elucidate some of the representational responsibilities of such formalisms.

1. A LOOK AT THE EVOLUTION OF SEMANTIC NETWORKS

The last ten years have seen a tremendous explosion in the number of efforts directed toward developing memory models that might be considered networks, and the literature has expanded to the point where only with extreme effort can one maintain familiarity with the entire field. To treat fairly all of the work that has led to our current state of knowledge about network knowledge representation would be a Herculean task, and one requiring far more space and time than is convenient here. Therefore, my analysis will begin with Quillian's [1966] work and will not discuss the many earlier efforts of Gestalt psychology, perception-by-reconstruction theories (especially Bartlett [1967] and Neisser [1967]), and artificial intelligence that have had significant effects on the current shape of semantic nets. I shall only briefly outline the various major contributions to the semantic network literature, and hope that the bibliography at the end of this chapter will provide sufficient direction for the reader more interested in historical trends and details on the representations sketched here. I shall not proceed strictly chronologically (many of these projects developed simultaneously), but shall instead broadly outline three major groups of work: the early nets that provided the basic structure, those which attempted to incorporate linguistic case structure, and several more recent important foundational studies.

1.1. The Early Nets

The idea of a semantic network representation for human knowledge is generally acknowledged to have originated in the work of Quillian [1966, 1967, 1968, 1969; Bell and Quillian, 1971]; Quillian proposed an association network model of "semantic memory" in his Ph.D. thesis in 1966. His intent was to capture in a formal representation the "objective" part of the meanings of words so that "humanlike use of those meanings" would be possible [1966, p. 1]. The representation was composed of nodes, interconnected by various kinds of associative links, and closely reflected the organization of an ordinary dictionary. The nodes were to be considered "word concepts," and links from a concept node pointed to other word concepts, which together made up a definition, just as dictionary definitions are constructed from sequences of words defined elsewhere in the same volume. The structure thus ultimately became an interwoven network of nodes and links.

In Quillian's structure, each word concept node was considered to be the head of a "plane," which held its definition. Figure 1 [Quillian, 1968, p. 236] illustrates a set of three planes (indicated by solid boxes) for three senses of the word "plant." Pointers within the plane (the solid links in the figure) are those which form the structure of the definition. Quillian postulated a small set of these, which included subclass (e.g., the relationship of PLANT 2 to APPARATUS in the figure), modification (e.g., APPARATUS is modified by the USE structure), disjunction (labeled by OR), conjunction (labeled by AND), and subject/object (e.g., the parallel links from USE to PEOPLE (the subject) and to = A (the object)). Pointers leading outside the plane (the broken links in the figure) indicate other planes in which the referenced words are themselves defined. The fact that in Quillian's structure words used in definitions of other words had their own planes, which were pointed to by place-holder nodes within those definitions, corresponded to the important "type/token" distinction. Each word was defined in only one plane in the structure (the head of the plane being the "type" node), and all references to a word went through intermediate "token" nodes. Thus definitions were not repeated each time a word concept was referenced.

Quillian's desire of his semantic memory model was that it might serve as a general inferential representation for knowledge. He presented in his thesis several examples of an inference technique based on the notion of a spreading activation intersection search—given two words, possible relations between them might be inferred by an unguided, breadth-first search of the area surrounding the planes for words; this search was carried out by a propagation of some kind of activation signal through the network. A search would fan out through links from the original two planes to all planes pointed to by the originals, until a point of intersection was found. The paths from the source nodes to the point of contact of the two "spheres of activation" formed by the search would indicate a potential relationship between the two word concepts. Quillian hoped that in this way information input in one frame of reference might be used to answer questions asked in another. The use of information implicit in the memory, but not stated explicitly, was one of the important features of the memory model.

* The belief that properties of a node could be found by an expanding search led Quillian to the idea that a word concept's "full meaning" comprised everything that could be reached from the patriarchal type node (the head of its defining plane) by an exhaustive tracing process.
The networks as a data structure in an implemented computer-aided instruction program. The SCHOLAR program had a knowledge base that described in network terms the geography of South America. A student could participate in a “mixed-initiative” dialogue with the system, asking and being asked questions about the data base.

SCHOLAR’s data base made some important new contributions to Quillian’s nets. Carbonell began to distinguish “concept units” (like LATITUDE) from “example units” (like ARGENTINA), setting the stage for the later notion of instantiation. In addition, a notion of Quillian’s called tags was expanded and used extensively. Figure 4 [Carbonell, 1970b, p. 194] illustrates the SCHOLAR units for latitude and Argentina; in the text part of the figure, the name of a unit follows RPAQQ (a LISP value-setting function), and anything within the unit that follows a left parenthesis is an attribute. Tags on relations are indicated by parenthesized pairs following the attribute names e.g., the SUPERP of LATITUDE is LOCATION, and has the tag (1 2). The most important of the tags in SCHOLAR was the irrelevancy tag (l-tag), which could explicitly increase the semantic distance between two nodes. 1-tags were used to determine the relevance of certain facts in a given context, and allowed the system to start with the

* Instantiation has become one of the most well-known aspects of semantic net formalisms. The general idea is the association of a particular individual with the class of which it is a member, and in most notations, this is reflected by the construction of an individual description based on a generic description that the individual satisfies. Thus, while we primarily think of instances as things in the world that are manifestations of our abstract concepts, the term “instantiation” is very often used to refer to the production of a description of an individual based on a more general description. I shall later (Section 3) use the term “individualization” (of description) for this latter intent, avoiding the potential confusion over what the term “instance” really means.

* The reader is also referred to an interesting article by Collins and Quillian called “How to make a language user” [1972b], in which they summarize many of the things that they learned from their experiments about language and memory.
Part of the reason that certain properties could be inferred from such a memory was its use of a link indicating a "subclass" relationship and a link specifying a "modifies" relation. A concept could be defined in terms of a more general concept (of which it was a subclass) and a modifying property, which was a combination of an attribute and a particular value for that attribute." In this characterization, properties true of a class were assumed true of all of its subclasses, except for the modifications. As a result, the superclass chain extending upward from a concept embodied all of the properties true of that concept. Thus the semantic net represented the combination of two important types of memory feature—a superclass—subclass taxonomic hierarchy, and the description of properties (attribute/value pairs) for each class. Earlier work done by Lindsay (see Lindsay [1973] for a later discussion of Lindsay's original work) and Raphael [1978] can be seen to be the precursors of this important marriage.

Quillian later cleaned up his memory model a bit. He eliminated the type/token distinction by making everything in the net a pointer, and, in a project called the "Teachable Language Comprehender" (TLC) [1969], he investigated its utility as a knowledge base for the reading of text. In TLC, a property was formally defined to be an attribute (something relational concept), a value, and possibly some further "subproperties." Properties were used in the definitions of "units," which represented the concepts of objects, events, ideas, assertions, etc.: a unit was defined by its superset and a set of refining properties. For reading, an intersection technique was used to find relations between words encountered in a text (this was augmented by the application of certain "form tests" as syntax checks). Figure 2 [Quillian, 1969, p. 462] illustrates a simple unit. The unit being defined in this figure is the one for "client." The unit indicates that a CLIENT is a PERSON (i.e., PERSON is its superset), with a further qualification indicated by the second pointer from the unit to a restricting property. That property combines the attribute EMPLOY with a value PROFESSIONAL and the subproperty BY THE CLIENT.

While TLC was an interesting model for finding connections between word meanings, its success in reading was limited. TLC's failure to achieve understanding was at least in part due to its insufficient set of link types and the fact that the search did not take into account the meanings of the various links. Despite the many shortcomings of his model, however, Quillian's early papers contain the seeds of most of the important ideas that are today the mainstays of semantic nets.

Quillian's revised TLC format gave rise to two other important studies. With Collins, Quillian undertook a series of experiments to test the psychological plausibility of his network scheme [Collins and Quillian, 1969,
most relevant aspects of a unit when describing a concept to the student. In addition, SCHOLAR introduced temporary, time-dependent tags. Also, while SCHOLAR's units looked much like Quillian's TLC units, the properties associated with a unit had as their first elements the names of attributes, rather than pointers (resurrecting the type-token distinction). Thus the precedent was set for naming links—associating arbitrary labels with the associations between units. In addition to several special attributes (SUPERF for superconcept, SUPERP for superpart, and SUPERA for superattribute), things like LOCATION, TOPOGRAPHY, CITIES, and UNIT were now being encoded directly into the network. Another important precedent set in the SCHOLAR net was the intermixing of procedures with the declarative structure. LISP functions associated with units were used to actively infer properties that were not stated as declarative facts.

Another early effort, which proceeded independently of the Quillian/SCHOLAR work but made use of similar structures, was Winston's "structural descriptions" work at MIT [1970, 1975]. Winston created a program that could infer the “concept” of a physical structure such as an ARCH (see Fig. 5 [Winston, 1975, p. 198]), given encodings of a set of ex-

![Diagram](https://via.placeholder.com/150)

Fig. 4. SCHOLAR units. From Carbonell [1976b].

![Diagram](https://via.placeholder.com/150)


* While Carbonell claimed that no links were privileged [1970b, p. 112], that those like "SUPERF" are very special indeed is illustrated in Brachman [1979b].
amples of the structure in a network description language. The descriptions included nodes for concepts of physical objects (like BRICKS) in a scene, and labeled links representing physical relationships between the objects (e.g., LEFT-OF, SUPPORTED-BY). The interesting thing about Winston's networks (aside from the fact that he had actually written a program to induce generalizations from them) is that the relationships between concepts could themselves be modified or talked about as concepts. For example, in the very same notation, B could be described as LEFT-OF C, and LEFT-OF described as OPPOSITE RIGHT-OF. Winston also used the same language as his comparison language for determining differences between examples.

One problem with Winston's notation, as with each of the others mentioned so far, was its complete uniformity. While the notions of superconcept and instance were included in these nets, there was no acknowledgement of their difference from domain-specific notions like location and subject. One could not "see" a hierarchy by looking at the structure, and important notions like inheritance were obscured by an overly uniform mixture of domain-specific and general "properties." However, with the groundwork laid by Quillian, Collins, Carbonell, and Winston, almost all of the semantic net apparatus used in the 1970s is already accounted for, and very little has really changed since then (at least until very recently, as Section 3 will attempt to show).

1.2. Case Structures

The work of Fillmore on linguistic case structure [1968] helped focus network attention onto verbs. Those interested in processing natural language with semantic nets began to think of a sentence as a modality coupled with a proposition, where a modality captured information such as tense, mood, manner, and aspect, and a proposition was a verb and a set of filled-in cases. There were believed to be a reasonably small number of cases (i.e., relationships in which nominals could participate relative to the verb of a sentence), and several people set out to incorporate this belief in network formalisms. The fact that properties in semantic nets were clustered around nodes made the nodes ideal places to anchor cases—if a node were thought of as a verbal concept, its associated attribute/value pairs could easily be case/filler pairs.

Simmons et al. [1968], Simmons and Bruce [1971], Simmons and Slocum [1972], Simmons [1973], Hendrix et al. [1973] used this notion very early in work that developed from the older Protosynthex system. Simmons' networks became centered around verbal nodes, with pointers labeled with case names to the participants in the action represented by the node (see Fig. 6 [Simmons and Bruce, 1971, p. 525]—the verbal node here is Cl, a TOKen of the verb Make). The verbs themselves were grouped into "paradigms," according to the sets of case relations in which they participated.

Simmons' networks focused on the understanding and generation of particular sentences—nothing attention seems to have been given in the original work to the semantic network as a hierarchical classification device, nor to the place of general "world knowledge" in the overall scheme. Thus no classification of verbs, or nouns, for that matter, existed outside the similar case-frame grouping (the paradigms), and no definitions of general concepts seemed to exist at all. Recently, however, some sophistication has been added to these networks, including substantial use of superconcept and instance links. In addition, quantification and deductive mechanisms are discussed in Simmons and Chester [1977].

A similar incorporation of case structures into a network framework was achieved by Rumelhart et al. [1972], Norman [1972, 1973], Norman et al. [1975], and Rumelhart and Norman [1973]. Their attempt, spanning several years, included many of the features that Simmons had left out, although their orientation was more psychological and thus dealt with more aspects of memory. The Rumelhart et al. networks included nodes for concepts, nodes for events, and nodes for episodes—sequences of events clustered together. General definitions of concepts in the network were encoded in a straightforward manner, with case-like pointers indicating parts of nominal concepts and agents and objects of verbs, as illustrated in Fig. 7 [Rumelhart et al., 1972, p. 224]. Unfortunately, their notation was also very uniform, so that all links looked the same. In addition, the infamous IS-A link (see Woods [1975] and Cercone [1975]) was used to indicate type-token relations as well as subset relations, and many other relations were not motivated or explained—the English mnemonics are all that we have to indicate their semantics. Relatively little attention was given to the structure at the foundational, logical adequacy level, so that the inheritance relations between concepts were not always clear.

On the other hand, the Rumelhart and Norman group made an effort to account for procedural-type information directly in their notation (using a link called IS-WHEN), and integrated case-type information with other
“world knowledge.” They included definitional as well as instantiated (propositional) constructs, and, all in all, they captured many good ideas in their nets.

Another important piece of work that deserves at least brief mention here is Schank’s “conceptual dependency” representation [1972, 1973a,b; Schank et al., 1973]. While Schank himself does not seem to believe in semantic memory [1974, 1975], his conceptualizations very much resemble concepts in systems like Simmons’ and Rumelhart and Norman’s, as evidenced in Fig. 8 [Schank, 1973b, p. 201]. A conceptualization consists of a primitive act and some associated cases, like “instrument” and “direction.” In conceptual dependency diagrams, arrows with different shapes and labels indicate the case relations. For example, in Fig. 8 the R relation (a three-pronged arrow) indicates the recipient case, while the I relation indicates the instrument of the conceptualization. For example, the instrument of “the boy hit the window with a stone” is “the boy” and “the stone.”

Schank’s contribution to the study of knowledge representation, while controversial, is an important one. His cases are “deeper” than those of Simmons, and begin to attack knowledge structure at the primitive level. Conceptual dependency was incorporated as the memory structure of the Margie system, which was a natural language understanding system that could parse an input sentence into the deep conceptual structure and rephrase it in a number of different ways. Schank and Riegler [1974] developed some important inferential properties for their memory structures, and their work has had a great influence on much of the later work in this field. The reader should consult Wilks [1974] and Cerccone [1975] for two excellent expositions of Schank’s work.

In more recent work, Riegler has attempted to deal in greater depth with the relations between actions and states [1975, 1976, 1977; Riegler and Grinberg, 1977]. Commonsense Algorithms (CSAs) capture information of a much more dynamic sort than that handled by the traditional, static concept networks. Riegler has nodes that represent not only primitive actions, but states, statechanges, wants, and “tendencies” (a tendency in CSA representation is a kind of action that takes place without the effort of an intentional force; one such tendency, for example, is gravity). There is a small repertoire of primitive link types, which are used to represent the underlying dynamic relationships between the actions, states, etc. (“ten theoretical forms of inter-event causal interaction” [Riegler and Grinberg, 1977, p. 250]).

CSA links stand for relations like causality, enablement, concurrency, and the like, with the primary emphasis on expressing the cause and effect relationships that make physical systems work. While the notion that causality can be captured in a single link is debatable, CSAs may provide a useful way to express dynamic information that in other systems is supposedly captured by unstructured relational links, and may do so in a complete enough way to allow the simulation of certain physical mechanisms, like the reverse-trap flush toilet [Riegler, 1975] and the reasonably complex home gas forced-air furnace [Riegler and Grinberg, 1977].

Two other important treatments of memory with verb-centered case-like systems surfaced in the early 1970s. Heidorn’s thesis [1972] parlayed a simple hierarchical network and instantiation mechanism into a system (called NLPQ) which could “understand” a queuing problem described to it in English. From this description, NLPQ could produce both an English restatement of the problem and a complete program (written in GPSS) for simulating the situation described. By including in advance some simple case frame definitions of actions relevant to queuing situations (for example, “unload” takes an agent, a goal, a location, and a duration), Heidorn provided his system with a built-in definitional context for the description of a particular situation. During an initial conversation with
the user, the NLPQ system would build an "internal problem description (IPD)." This IPD comprised a set of instances connected appropriately to the general definitions (see Fig. 9 [Heidorn, 1974, p. 95]). NLPQ could consult those definitions and tell when the problem description was incomplete; it could thus intelligently ask the user for missing information. Although Heidorn's network was very simple and uniform (it was not very deep, concepts had very simple structure, and the SUP link was used for both subconcepts and instances), he achieved a rather dazzling effect by incorporating it in a general grammar-rule language and by starting with a set of concepts well-matched to the simulation language in which the output was produced.

The other "case" study produced a strongly psychologically oriented memory structure called HAM (for Human Associative Memory) [Anderson and Bower, 1973, 1974]. The elements of HAM were propositions, binary trees that represented the underlying structure of sentences. A simple proposition of this sort is depicted in Fig. 10 [Anderson and Bower, 1973, p. 165]. Relations allowed between nodes in the trees included set membership (the E links in Fig. 10) and subset, some cases like subject (S in Fig. 10), object (O), location (L), and time (T), and some logical indicators like predicate (P), context (C), and fact (F)—all represented uniformly. Propositions in HAM had truth values, and were supposed to convey assertions about the world; Anderson and Bower's notation failed to account for the internal structure of nominal entities. There were many problems with this simple notation, some of which are discussed in Schubert [1976], a work whose detail on the logical structure of semantic networks in terms of predicates and propositions makes it clear that HAM's propositional notation is insufficient. However, Anderson and Bower produced an extensive investigation into the state of the relevant philosophical and scientific work at the time of their own work, and their detailed psychological discussions should be consulted. Although their model is admitted to be inadequate and the semantics of their representation is not thoroughly worked out, their book is a milestone of start-to-finish research in a field often plagued by less than thorough work.

1.3. Concern for the Foundations

Unfortunately, most of the early work covered above suffers from a lack of explicit acknowledgement of some fundamental principles of knowledge representation design. Authors are most often intuitive when describing the semantics of their representations,* and as the network no-

* For example, "Intuitively, the nodes in the tree represent ideas and the links relations or associations between the ideas" [Anderson and Bower, 1973, p. 139]. "In this system a large part of the information is about the words and concepts of the relevant domain of discourse..." [Heidorn, 1972, p. 35].
tations get more complex, more and more of the assumptions are left to the reader’s imagination. Most of the early representations were not extensible in a general way (i.e., the system designer must intervene to add new case relations), and in general, the combination of set operations and descriptive concept operations that the semantic net is based upon has been poorly understood (see Brachman [1978b], especially Chapter 4, for details). All of the notions mentioned so far are seductively uniform—conceptual relations (e.g., agent, color, left-of) and underlying knowledge mechanisms (e.g., superset, “iswhen,” member) are expressed in indistinguishable terms. In Section 2, I shall contend that this homogeneity is misguided and confusing.

However, in addition to that described in Section 3, some recent efforts have set out to remedy this inadequacy. Among the more important of the earlier and concurrent projects that attempted to deal with the expressive inadequacy of semantic nets are the work of Cercone and Schubert at the University of Alberta, and the work of Levesque and Mylopoulos at the University of Toronto, to which we turn in a moment. Several years earlier, however, Shapiro [1971a,b] introduced the important distinction between the “item,” or conceptual level of network, and the “system” level—the structural level of interconnection that ties structured assertions of facts to items participating in those facts (i.e., indicates bindings). System relations are the labeled links in the network, and their semantics is determined by the set of processing routines that operate on them. Item relations are concepts that happen to be relational in nature, and are represented by nodes (items) just as are other, nonrelational concepts. Thus, a relationship like LOVES would appear not as a link in the net, but as a node. Particular assertions of the relationship would also be nodes, with AGENT and OBJECT system links to nodes for the participants, and a VERB link back to the node for LOVES (see Fig. 11 [Shapiro, 1971a, p. 43]). In Fig. 11, the top three nodes are assertions of particular LOVES relationships. Shapiro makes no suggestion as to how the general verb itself should be defined in network terms (that is, what makes a concept LOVES as opposed to any other verb with a similar case frame).

Shapiro’s distinction explicitly separates underlying primitive cases from all other (conceptual) relations. He also explains how rules for deduction can be encoded directly in his formalism, and discusses at length a language for doing retrieval from his network structure. His early work gives us no guidelines for what the set of system relations should be (his examples suggest linguistic cases), nor does he talk about the semantics of items, except to imply through his search mechanism that sets are important. Shapiro’s claim is only that what he has given us is an epistemologically neutral structure, a general language on top of which many models of knowledge might be constructed. This in itself, however, represents a significant advance over previous networks in the distillation of two very different levels of representation.

Between the time of Shapiro’s thesis [1971a] and the more recent work to which I have alluded, others have tried to resolve some of the inadequacies of the homogeneous standard evolved from Quillian’s Semantic Memory. Hays [1973a,b], in his “cognitive networks,” has attempted to differentiate some of the semantics of network notations, and to be more formal than earlier authors about network structures (he specifies four node types, including “modalities,” and five major link types). Among other things, his work has contributed the distinction between a “manifestation” of an object and an “instance.”

Hendrix [1975a,b, 1976], in attempting to provide an adequate quantification mechanism for semantic network concepts, introduced what has become a very broadly utilized facility—partitions,† or formal

* Objects in Hays’ epistemology are permanent. However, they do change over time (e.g., a person is at various times an infant, a child, an adolescent, an adult, etc.). Manifestations are different concepts of the same object at different places or stages of its existence.

† Scragg [1975] has, apparently independently, introduced a very similar mechanism, which he calls “planes.”
groupings of concept nodes. Figure 12 [Hendrix, 1975a, p. 239] illustrates the use of partitions (indicated by rectangular dashed boxes) to represent “Every city has a dogcatcher who has been bitten by every dog in town.” In this figure, the two larger “spaces” hold the scopes of the universal quantifiers: the “form” link points to a space representing the scope of the universally quantified variable, which is encoded by a node pointed to by a “for all v” link. The node labeled “p” is an implicitly existentially quantified node, representing the particular dogcatcher for any given town.

Partitioning has many potential uses; for example, it can be used to provide a context mechanism, whereby entire areas of memory may be opened up or sealed off at relevant times (this allows reasonable groupings of beliefs). It should be pointed out that the nodes in many of Hendrix’s nets represent sets as well as “prototypes”, and the introduction of case-like properties for concept nodes makes them susceptible to the same confusions as all of the older, uniform nets (this is evidenced by relations like “creature” and “assailant” being directly encoded as links in his nets). Apparently, however, different space-types are used to distinguish different uses of the same link, and the nonlogical links are not really primitive in the system—they are introduced by “delineations” associated with general verbal concepts like OWNINGS. This is not obvious in some of the earlier papers, but see Hendrix (this volume) for the supporting details.

Partitions have become a mainstay of many recent semantic nets and are an indisputably helpful mechanism for representing higher level phenomena like quantification, context, and structural “plots” [Gross, 1977]. When viewed as a mechanism, with no epistemological claims about their expressive adequacy (which depend on each individual’s use of them), partitions do not come under the jurisdiction of our criticisms in Section 2. When partitions implement mixed sets of relationships (like “creature” and subset), then they are open to the kind of complaint we shall lodge in that section. That is, each partition (space) type used in a system is open to its own epistemological constraints, just as is each use of the simple, general notion of a node.

In 1975, a very important paper by Woods appeared; this study of “what’s in a link” for the first time seriously challenged the logical adequacy of previous semantic network notations [Woods, 1975]. Woods pointed out the intensional nature of many of the things we call upon nets to represent (see also Chapter 5 of Brachman [1978b]), and discussed in detail several important challenges for network notations that had not been previously acknowledged, let alone successfully met. We were asked to begin to consider the semantics of the representation itself, and to be held accountable for things previously brushed aside under the auspices of “intuition.” The work to be described in Section 3 is, to some extent, a broader and deeper investigation in the same spirit as the Woods paper, a continuation of the semantic investigative work only begun there. It is hoped that many of Woods’ challenges have been overcome by the structures illustrated in that section and in [Brachman, 1978b].

Some of the issues raised by Woods—the more logically oriented ones—have been recently treated in a series of papers by Cercone and Schubert [1978], Cercone [1975], and Schubert [1976]. In their attempts to extend the expressive power of network notation, Schubert and Cercone have expended considerable effort in the investigation of the underlying logical content of the node-plus-link formalism. Many of the issues of knowledge representation that were emphasized in my thesis [Brachman, 1978b] were raised in various papers from Alberta; in particular, an excellent criticism of the naive notion of the existence of a small number of “conceptually primitive relations” (i.e., cases) reflects a similar intuition about roles (see Section 5.1.3.1 of Brachman [1978b], Schubert [1976, pp. 168–170], and Cercone [1975, pp. 79–80]).

The notation developed by Schubert and Cercone is propositional—an important basic node type in the network is the predicative concept node, which is instantiated by conjoining at a proposition node a pointer to the predicate and a pointer to each argument of the predicate (see Fig. 13 [Cercone, 1975, p. 56]). The links used are all predefined system links, used only to point out the particular predicate invoked and to order the arguments. All of the conceptual work is done by the particular predicates pointed to with PRED links from the proposition nodes. Schubert and Cercone claim also to have concept nodes for individuals and sets, although it is not clear from the notation where these interpretations are expected. Given the propositional nature of the notation, a series of logical connectives and quantification conventions can be unambiguously (and explicitly) represented. In addition, Schubert and Cercone provide facilities for lambda-abstraction and various other intensional operations, and include time primitives for certain types of predicates. Schubert [1976] discusses
the clear correspondence of his notation to predicate calculus, providing for the first time a clear standard of reference for network (logical) adequacy.*

While the work of Cercone and Schubert begins to answer some of the questions raised in Woods' paper, theirs is still only a neutral logical language. This notation, as all others discussed so far, offers no guidelines to its users on how to structure concepts in terms of the primitives of the notation. The language is as general, uniform, and low-level as predicate calculus and is up to the designer of the particular network how to structure his world in terms of predicates and propositions. While Schubert's notation unambiguously accounts for many of the underlying logical operations of the semantic network, something more seems to be needed for it to be a truly useful representation of knowledge. This seems to involve looking at network structures at a slightly "higher" level.

Some hints on higher level primitives have been afforded us by some more recent efforts in network formalisms. Fahlman [1977] has designed a network system comprising two major parts: a parallel memory scheme, which allows propagation of markers through a network composed of special-purpose hardware, and a language (called NETL) for representing knowledge on top of the parallel memory. There are several important things to note about Fahlman's work. His is perhaps the first attempt to account for network-implementing hardware in its own right. The marker propagation and detection mechanism eliminates much of the costly search intrinsic to previous, nonparallel systems. Further, he introduces the idea of a "virtual copy" as a dominant organizing concept. This is a convenient way to think about inheritance in semantic nets, since it lets us assume that all properties at a parent node are (virtually) available at its subnodes. When a real copy is needed, as, for instance, when a property is to be explicitly modified, Fahlman has us create a MAP-node. The parallel-processing scheme makes virtual copy and map links act as short circuits in the appropriate circumstances, thereby allowing any inherited definitions to be immediately available.

Further, Fahlman introduces the "role" as a type of individual, whose universe of existence is another concept. While he at times, I believe, confuses the notion of a functional role (like AGENT) with that of a role filler (like PERSON), he seems to be on the right track in terms of the structure of concepts. In my own work (see Section 3 here), I have found this role notion to be critical, and have what amount to MAP-nodes also. A good deal of Fahlman's foundations could be used to support other network schemes.

Role-nodes as parts of structured descriptions also constitute a critical element in the work of Philip Hayes [1977a,b]. Hayes' networks have two levels of structure, just as those to be presented in Section 3 have: the internal structure of "depiictions" (concepts), and relationships between depictions as wholes. Briefly, a depiction expresses the structure of an entity through a set of PART-OF and CONNECTED relationships between other entities that make up its parts. For example, in Fig. 14 [Philip Hayes, 1977a, p. 93], the depiction D-HUMAN (indicated by dashed lines) partially expresses the structure of a human (represented by the node N-HUMAN) in terms of an ARM and a TORSO. In the depiction D-HUMAN, N-ARM acts as a depictor; at the same time, in D-ARM, N-ARM is the depictee—the subject of the depiction.* Thus, while it is a thing unto itself in one structure, it acts as the specifier of a role to be filled in another. In some cases, Hayes contends (and I concur), the role can only exist within the larger context. For example, an arm cannot exist without implying the existence of some human; in that case, N-ARM would be an SNMP for D-HUMAN, and the dependency would be expressed in an SNMP structure (for sine qua non) involving D-ARM and D-HUMAN.

* While N-ARM is the same node in both depictions, links to it are only "visible" from the depiction from which it is viewed. That way various uses of ARM from more than one context can be kept distinct.
While Hayes does not distinguish the role itself from the role filler: (see Section 3), and CONNECTED is much too simplistic to capture relations between roles, the very fact that Hayes has roles at all is significant. Concept structure involving roles is strictly enforced in instantiation, using a structure called a "binder." In Fig. 14, there are two binders (indicated by the rectangular boxes, the arrows coming in to them, and the dots at intersections), representing Fred and Fred's arm. The binder captures the fact that roles are inherited as part of a structure. There are explicit connections between role definitions (in the deictions) and role filler/instance pairs (in the binders), just as I propose in Section 3 (although the exact nature of the relationships is not spelled out in Hayes' structure). The explicit acknowledgement of these relationships is a very important development in the history of semantic networks.

Finally, a joint concern for higher-level (nonlogical) structures and their semantics in a semantic network formalism has surfaced in the work of Levesque and Mylopoulos at Toronto [in this volume; Levesque, 1977]. Their efforts attempt to provide a procedural semantics for the relations in a network by associating with a class (concept) a set of four operations: add an instance, remove an instance, fetch all instances of the class, and test for being an instance of the class. Classes are given internal structure with slots; parts fill these slots, generating a "PART-OF hierarchy." The classes themselves are organized in a "IS-A hierarchy," which expresses generalization relationships between classes and subclasses.

In addition to these two hierarchies, the system of Levesque and Mylopoulos also has an "instance hierarchy." Every class is itself an instance of the class CLASS, which is termed a "meta-class." Adding this distinction allows a precise account of inheritance and of relations often mistaken in more uniform schemes—including the descriptions of the programs themselves. Levesque and Mylopoulos also provide (in this volume) nice accounts of the distinctions between structural and assertional properties and between property attributes and property values, and account with their procedures for the interdependencies between pieces of a structure. As such, their account would provide a good set of tools for exploring the semantics of the representation to be presented in Section 3. The only major shortcoming is the lack of an explicit representation of the relationships between the parts of a class, since their dependencies are only implicitly accounted for in the four programs associated with a class definition.

2. "ONE MAN'S CEILING IS ANOTHER MAN'S FLOOR"

Given this rich and interesting history, what can we conclude, is a "semantic net"? About the only thing in common among all of the above-discussed formalisms is their connectivity—the fact that they all claim to be made of links and nodes. But these two kinds of entity are really just descriptive of implementations—they have nothing to say about the epistemological import of these networks. While Hendrix states, "Broadly speaking, any representation interlinking nodes with arcs could be called a semantic network ..." [Hendrix, 1977, p. 984], I believe that there is something amiss in this interpretation. Hendrix's definition is indistinguishable from that of a graph—and there is nothing inherently "semantic" about a graph.

The "semanticness" of semantic nets lies in their being used in attempts to represent the semantics of English words. Besides nodes and links, the common thread that has held together the many efforts described in Section 1 is the fact that most of the networks were to be used in understanding natural language. Semantic nets have become a popular meaning representation for use in natural language understanding systems.

Despite the fact that virtually all semantic networks have been used to represent the "concepts" corresponding to words and sentences, there has been very little agreement on how to factor such knowledge. The most important disagreement—as evidenced by the fact that we find no two sets of links the same in the literature—has to do with the structural decomposition of a concept. While only the most recent work [Brachman, 1978a,b; Levesque and Mylopoulos, in this volume; Smith, 1978] has dealt with concept-structuring per se, every network implicitly embodies a theory of the primitive elements that make up a concept. Structural links holding together the arguments of a logical predicate, "deep semantic cases," and even conceptual relations that are supposed to exist between objects in the world have all been proposed as semantic network links.

Consider the following statements:

(1) The distinction between these alternatives appears to be a significant one since logical forms are clearly formal languages within which meanings of surface strings are represented, whereas the latter are labelled graphs which somehow represent these same meanings. This distinction quickly evaporates, however, the moment one observes that a network is basically a particular choice of representation (at the implementation level) for some (conceptual level) logical form. [Nash-Webber and Reiter, 1977, p. 121]

(2) If someone argues for the superiority of semantic networks over logic, he must be referring to some other property of the former than their meaning ... [Pat Hayes, 1977, p. 561]

(3) A semantic network purports to represent concepts expressed by natural-language words and phrases as nodes connected to other such concepts by a particular set of arcs called semantic relations. Primitive concepts in this system of semantic networks are wordsense meanings. Primitive semantic relations are those that the verb
of a sentence has with its subject, object, and prepositional phrase arguments in addition to those that underlie common lexical, classificational and modificational relations. [Simmons, 1973, p. 63]

How are we to rationalize such disparate views of what has always seemed to be a single formalism?

2.1. Will the Real Semantic Network Please Stand Up?

The key issue here is the isolation of the primitives for semantic network languages. The primitives of a network language are those things that the interpreter is programmed in advance to understand, and that are not usually represented in the network language itself. While there is, of course, no one set of primitives that is the set, for any single language there should be one fixed group of primitive elements. Only with a fixed set of primitives can a fixed interpreter be constructed. It would be difficult to justify an interpreter in which the set of primitives changed meaning, or for which it was expected that new primitives were to be added in the course of interpretation.

The view I shall take here is that the history of semantic nets and their utility as a representational device can best be understood by carefully examining the nature of the primitives that have been proposed. Since the semantics of any given language is dependent on the interpretation of the primitive elements and a set of rules for combining them into nonprimitive elements, the "well-definedness" of a network language rests heavily on the set of node and link types that it provides. While it is difficult to make simple, clear-cut comparisons of primitives from one language to the next, it is possible to distill a small number of distinctive types of nodes and links from the formalisms discussed in Section 1. Each of these types can be considered to form a consistent conceptual "level". I shall therefore propose a set of levels, or viewpoints, with which to understand the range of semantic network primitives that have been used in the past. Each of the levels will have a set of link types whose import is clearly distinct from those of other levels.

Any network notation could be analyzed in terms of any of the levels, since they do not really have any absolute, independent "existence." For example, a particular concept might be structured with semantic cases. These cases can be understood as sets of logical predicates, which, in turn are implemented with some set of atoms and pointers. However, each network scheme does propose an explicit set of primitives as part of the language, and this is the set that is actually supported by the language's interpreter. The particular sets of primitives that are proposed in particular languages are the ones of interest to us here. Understanding past problems with semantic nets and what one is getting when one uses a particular semantic net scheme are both facilitated by looking closely at these explicit primitive sets. As we shall see, one set of difficulties has arisen because most network notations freely intermix primitives of different types.

The diverse opinions on the primitives of semantic networks expressed both explicitly in the literature and implicitly in the style of existing networks indicate that there are at least four main levels of primitive to be considered. The first view, expressed by the quote from Nash-Webber and Reiter above, considers a semantic network to be an implementational mechanism for a higher-level logical language. This view we might call the implementational form of semantic nets. In implementation level networks, links are merely pointers, and nodes are simply destinations for links. These primitives make no important substantive claims about the structure of knowledge, since this level takes a network to be only a data structure out of which to build logical forms. While a useful data-organizing technique, this kind of network gives us no more hint about how to represent (factor) knowledge than do list structures.

A second view sees a semantic network as a set of logical primitives with a structured index over those primitives. In this type of logical level net, nodes represent predicates and propositions. Links represent logical relationships, like AND, SUBSET, and THERE-EXISTS. The above quote from Pat Hayes expresses a viewpoint that is essentially the same as those implicitly expressed in networks by Schubert and Hendrix, to some extent Shapiro, and to a lesser extent Woods. In this point of view, logical adequacy, including quantificalional apparatus, is taken to be the responsibility of semantic network representations of knowledge. The aforementioned efforts express a tacit dependence on predicate calculus for knowledge factoring and espouse network schemes as useful organizing principles over normally nonindexed predicate calculus statements. In doing so, they make at least some claim about how knowledge can be meaningfully factored. It is interesting to note that almost all of the "foundational" work on semantic networks has been done at this logical level of interpretation.

The most prevalent view of networks is reflected in Simmons' above statement and almost all of the work discussed in Section 1. This is, in some sense, the "real" semantic net—a network structure whose primit-

* It is probably also desirable to have this set as small as possible. However, at the current stage, we shall settle for any set that is adequate. One of the purposes of this chapter (see Section 3) is to begin to circumscribe what would be an adequate set of primitives, and therefore, an adequate interpreter.

* The assignment of a notation to a particular level should not be taken as a value judgment (very few formalisms can, in fact, be assigned to a single level). Different level notations are useful for different tasks here. It is asked only that one become aware of the level at which one stops decomposing concepts, and understands the meaning of one's primitives as potentially decomposed into "lower" level ones.
tives are word-senses and case relations. As Simmons notes, in these conceptual level nets, links represent semantic or conceptual relationships. Networks at this level can be characterized as having small sets of language-independent conceptual elements (namely, primitive object- and action-types) and conceptually primitive relationships (i.e., "deep cases") out of which all expressible concepts can be constructed. The strongest view at this level is that of Schank and followers, which picks a small set of act types (e.g., GRASP, INGEST, PTRANS) and cases (e.g., INSTRUMENT, RECIPIENT) and claims that this set is adequate to represent the knowledge expressed in any natural language. Weaker views are embodied in the Norman and Simsons nets, where it is difficult to tell if there are any truly primitive knowledge pieces. In these nets, the belief in cases, and the particular sets settled upon, seem to be the unchanging elemental units.

Finally, going one step higher, we might consider networks whose primitive elements are language-specific. The only formalism that I know of at the current time that embodies this view is OWL, whose elements are expressions based on English. In such a formalism, one would presumably "take seriously the Whorfian hypothesis that a person's language plays a key role in determining his model of the world and thus in structuring his thought" [Martin, 1977, p. 985]. In OWL there is a basic concept-structuring scheme (see Hawkins [1973]) that is used to build expressions, and strictly speaking, the principles of "specialization," "attachment," and "reference" are the primitives of the language. However, these primitives are neutral enough to be considered implementational, and thus the knowledge itself can be considered to form the structure of the database. This seems operationally reasonable when OWL is looked at in detail—the two expressions (HYDRANT FIRE) and (MAN FIRE), while both specialized by FIRE, can have the specializations "mean" different things based on the rest of the network structure. This linguistic level represents perhaps the most radical view of semantic nets, in that the "primitives" are language-dependent, and are expected to change in meaning as the network grows. Links in linguistic level networks stand for arbitrary relationships that exist in the world being represented.

It should be obvious that each of the above viewpoints implies a set of primitive relationships substantially different from the others. Relationships between predicates and propositions (e.g., AND, ARGUMENT1, PRED) are distinctly different than those between verbs and associated "cases" (e.g., AGENT, INSTRUMENT, RECIPIENT). Both of these are not the same as arbitrary relationships between things in the world (i.e., relations between the entities that the concepts are supposed to denote, e.g., COLOR, HIT). And further, none of these is the same as the relations between the parts of an intensional description, to which we now turn.

2.2. The Missing Level

While this characterization of the levels of semantic network representations covers virtually all of the work that has been done, there appears to be at least one level missing from the analysis. That this is the case is suggested by some of the more recent phenomena appearing in network languages, including "partitions," "deletions" [Hendrix, 1975a,b, 1976], and "binders" [Philip Hayes, 1977a,b].* These features suggest the possibility of organizations of conceptual knowledge into units more structured than simple nodes and links or predicates and propositions, and the possibility of processing over larger units than single network links. The predominant use of concepts as intensional descriptions of objects in a world also hints that there is a class of relationship that is not accounted for by the four levels already discussed. This kind of relationship relates the parts of an intension [Carnap, 1947] to the intension as a whole, and one intension to another. Intensional descriptions can be related directly by virtue of their internal structures, or indirectly by virtue of their corresponding extensions.* In addition, even the single most common trait of semantic networks—"inheritance"—suggests a level of knowledge structure between the logical and conceptual ones described above. Inheritance of properties is not a logical primitive; on the other hand, it is a mechanism assumed by almost all conceptual level nets, but not accounted for as a "semantic" (deep case) relation.

There must be some intermediate level at which a precise formal account of such notions can be constructed. The attempt to give conceptual units more structure than that of uniform configurations of links hints that at least some network designers have been thinking about "concepts" as formal objects, with predetermined internal organization that is more sophisticated than sets of cases. The formal structure of conceptual units and their interrelationships as conceptual units (independent of any knowledge expressed therein) forms what could be called an epistemology. I shall propose, then, an intermediate level of network primitive that embodies this formal structuring. This will be called the epistemological level, and it lies between the logical and conceptual levels.

The epistemological level of semantic network permits the formal def-

---

* In general, a characteristic of this level is that it should be language-independent.
+ "We have taken English as the basis for our knowledge representation formalism" [Slovovits et al., 1977, p. 2]. Not surprisingly, the view of the OWL group is that their "Linguistic Memory System" is a semantic network: "The most novel aspects of OWL are the structure of the semantic net (Hawkinson, 1975) . . . ." [Martin, 1977, p. 985]

* Nonnetwork languages, like KRL [Bobrow and Winograd, 1977], have similar types of mechanisms playing a more and more prominent part.
+ KRL has also focused intensively on the issue of description—most notably, what constitutes a description, and how descriptions in the above sense are inherently partial.
inition of knowledge-structuring primitives, rather than particular knowledge primitives (as in the Schank networks). Note that networks at the next higher level (conceptual) take as their primitives pieces of semantic knowledge and cases, but with no explicit accounting for their internal structure. While there is no universal agreement on what cases there are, everyone agrees that there are probably at least some cases, and they all seem to have a feel for what a case is. The basis for this agreement on the concept of case (or “slot”) and the inheritance of cases in a network is provided by the epistemological level, which explains the meaning of cases in general and provides a defining mechanism for particular ones.

In Section 3, I shall touch on some of the other operations that nets built out of explicit epistemological primitives should account for. In that section, I shall also introduce a formalism expressly based on those notions. Briefly, relations at the epistemological level include those used to structure intensional descriptions and those used to interrelate them. The former involves piecing together aspects of a description, including descriptions of the conceptual subpieces of an object and how they intertwine. One such type of conceptual subpiece is a case, the meaning of which is taken to be something built out of epistemological primitives. The latter type of relation specifies inheritance of subdescriptions between descriptions.

Table I summarizes our discussion of the five levels. The examples listed with the levels are suggestive of the philosophy of those levels; none is really a “pure” example of a single primitive type. Although a desirable goal (as we shall see below), it is not clear that a pure network at any level is attainable. Our task here, then, is to understand as well as possible each type of primitive so that the semantics of any formalism can be clearly and completely specified, even if it mixes elements of more than one level.

### 2.3. Neutrality, Adequacy, and Semantics

Despite the fact that we have isolated five distinct types of semantic nets, there are some universal notions that can be applied equally well to each type of network. These are neutrality, adequacy, and semantics. Each of these can be used as a criterion for judging the utility and formality of a given network language. It is desirable that a formalism be as pure as possible, adequate to handle its appointed representation task, and have a clean, explicitly specified semantics.

A network implemented at some level should be neutral toward the level above it. For example, logical nets are “epistemologically neutral,” in that they do not force any choice of epistemological primitives on the language user. Making “concepts” in logical nets, then, is a mixing of levels. Conceptual level nets must support many different linguistic systems and should be linguistically neutral (as Schank puts it, “the principal problem that we shall address here is the representation of meaning in an unambiguous language-free manner” [1973b, p. 187]). By the same token, of course, epistemological formalisms must be neutral in regard to particular semantic relationships. It is the job of the epistemological formalism to provide case-defining facilities—not particular cases.

A formalism at any of the four lower levels that is neutral toward the one above is a useful tool for designers of those at higher levels. Epistemological neutrality, for example, ensures flexibility in the design and definition of particular cases or nonstandard types of inheritance. It should be clear, then, that one of the main problems with many of the older formalisms was their lack of a clear notion of what level they were designed for. Almost universally, semantic networks have mixed primitives from more than one level (for example, particular cases were links in the same systems in which set membership was a link). In terms of neutrality, these formalisms were all less flexible than they could have been in serving knowledge bases designers who were building structures on top of them. Decisions were forced at more than one level at a time; the simultaneous freedom on some issues and lack of flexibility on others (at the same level) has been a constant source of confusion for network language users throughout the history of semantic nets.

Any level network can also be judged on its adequacy for supporting the level above it. If a semantic net can somehow support all possible linguistic systems of knowledge, then it has achieved conceptual adequacy. While the particular features of conceptual adequacy are open to debate, the notion of adequacy for the logical level is well understood (see Shubert [1976], for example). At the current time, it is less clear

<table>
<thead>
<tr>
<th>Level</th>
<th>Primitives</th>
<th>Examples (nonexclusive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementational</td>
<td>Atoms, pointers, propositions, predicates, logical operators</td>
<td>Data structures [Schubert, 1976] [Cercone, 1975]</td>
</tr>
<tr>
<td>Logical</td>
<td>Concept types, conceptual subpieces, inheritance and structuring relations</td>
<td>Data structures [Hendrix, 1975a,b] [Brachman, 1978a,b]</td>
</tr>
<tr>
<td>Epistemological</td>
<td>Semantic or conceptual relations (cases), primitive objects and actions</td>
<td>Data structures [Schank, 1972] [Simmons, 1973] [Norman et al., 1975]</td>
</tr>
<tr>
<td>Conceptual</td>
<td>Arbitrary concepts, words, expressions</td>
<td>Data structures [Szolovits et al., 1977]</td>
</tr>
</tbody>
</table>
what it would take for a network representation language to be epistemologically adequate. This is a subject (as, for example, treated in Brachman [1978b]) that is ripe for study, and to which I would like to draw the reader’s attention. Treatments of criteria for epistemological adequacy have heretofore been missing from network studies. Yet it seems that the understanding of knowledge representation languages in general will depend intimately on an understanding of what the elements of epistemological adequacy are, and how well given languages handle them. We shall look at some aspects of this in the next section.

Thinking in terms of adequacy gives us another reason why previous semantic networks have been difficult to assess. Given networks that mixed levels of primitives, it was impossible to tell what exactly the networks were adequate for. The recent push toward completely logical networks was in part motivated by the desire to achieve for the first time a network that was demonstrably adequate in a well-understood way.

Finally, each type of network language must be held accountable for its semantics—what, formally, do each of its elements mean, and what are the legal operations on them? In this respect, the attempts to define logically adequate nets (Schubert, Cercone, Hendrix, etc.) have a clear advantage over all of the others: once a mapping to predicate calculus is established, a formal semantics (i.e., Tarskian truth-theoretic) is available, essentially for free. This requirement, on the other hand, makes a formal semantics for linguistic level nets almost impossible to achieve, since it would require a formal semantics to be defined for the particular natural language involved. For conceptual level nets, only Schank and Rieger [1974] have provided a well-defined semantics, in that they have, in advance, specified sets of “inferences” for each of their predetermined primitive acts. Thus an act is defined in terms of its inferences, and there are rules for combining inferences into interpretations of larger, nonprimitive structures, based only on the primitives out of which they are built. Other conceptual level nets do not have fixed primitives, thereby making it difficult to provide an acceptable semantics.

Formal semantics for epistemological level languages are currently under study; studies of such semantics must however be done in parallel with those attacking epistemological adequacy, since the nature of the epistemological primitives is not yet understood, which therefore makes it hard to define a semantics. Three particular studies are of note here:

1. In Brachman [1978b], I attempt to ferret out the meaning of a network language by making each basic relationship available as a link, and then explaining in detail the epistemological significance of each link.

2. The work of Levesque and Mylopoulos at Toronto [in this volume; Levesque, 1977] investigates a procedural semantics in a similar manner—except that the nature of the procedures themselves is being dealt with in a general, network-expressible way.

3. Smith [1978] is working on a comprehensive paradigm for knowledge representation languages in general, which includes a noncircular explanation for the interpreter of the procedures, as well as accounts of “meta-description,” structured inheritance, believing as an active process that denotes, etc.

3. AN EPISTEMOLOGICALLY EXPLICIT REPRESENTATION

In this final section, I hope to illustrate what a network formalism at the epistemological level of knowledge representation should be concerned with. Such a formalism should have a clean, well-understood semantics, should not depend in any way upon the domain to which it is applied, and should adequately handle the significant epistemological relationships of concept structuring, attribute/value inheritance, multiple description, etc.

In order to make the ideas more concrete, I shall discuss epistemological primitives in terms of a particular type of formalism called Structured Inheritance Networks (SI-Nets) [Brachman, 1978a,b]. SI-Nets were developed expressly to address the above cited epistemological issues and to provide a useful explanatory tool for semantic level languages. SI-Nets constitute a class of network languages whose links represent epistemologically primitive relations. For the purposes of this discussion, we shall use a paradigmatic example of this class, called KLONE. While KLONE will be discussed only briefly here, details are available in Brachman [1978a,b].

The basic elements of KLONE (as they are in most semantic net schemes) are Concepts. Concepts are formal objects used to represent objects, attributes, and relationships of the domain being modeled. A Concept is thought of as representing an intensional object, and no Concepts

* The reader should be warned here that I am using the term “semantics” in its currently popular AI sense, wherein the meaning of a primitive is provided by the programs that operate on it. While the notion of links being meaningful by virtue of the programs that use them seems intuitively clear and reasonably precise, there is a lot more to be said on the issue. Semantics deals with the relationship between a symbol and what it denotes. Therefore not only should we take careful account of what here is a symbol (e.g., some marks on a piece of paper, an arrow with a word next to it, a set of bits in a computer), but we must also be precise about what these symbols denote (i.e., what are “epistemological relations” anyway?). Smith treats these problems in insightful depth in his Master’s thesis [1978]. See also Fodor [1978] for a recent critique of “procedural semantics.”
are used to represent directly extensional (world) objects. There are two main Concept types—generic and individual. Generic Concepts represent classes of individuals by describing the characteristics of a prototypical member of the class. Individual Concepts represent individual objects, relationships, etc., by individuating more general Concepts. Individuation is a relationship between Concepts. The term “instance” has been used in many network models to refer to an individuating description, as well as to the thing in the world that the individual description describes. Here, however, we shall use “instantiation” only as a relationship between a thing in the world and a Generic Concept, and not as a relationship between Concepts. Thus, the Arc de Triomphe (i.e., the one in Paris) is an “instance” of the Concept ARCH; the Individual Concept (call it ARC-DE-TRIOMPHE) that denotes the Arc de Triomphe “individuates” the Concept ARCH. The relationship between ARC-DE-TRIOMPHE and the real Arc is “denotation.” See Fig. 15 for a schematic picture of this three-way relationship.

3.1. Internal Concept Structure

The key observation of SI-Nets is that objects in the world have complex relational structure—they cannot, in general, be usefully taken as atomic entities or mere lists of properties (see Brachman [1977, 1978b] for detailed justification). A Concept must therefore account for this internal structure as well as for the object as a wholeistic entity. The KLONE formal entities that support this view of structured objects are Role/Filler Descriptions (Roles) and Structural Descriptions (SDs). Roles represent the conceptual subpieces of an entity, while SDs account for the structuring relationships between them.

The Roles represent the various kinds of attributes, parts, etc., that things in the world are considered to “have.” These include, for example, such things as parts (e.g., fingers of a hand), inherent attributes of objects and substances (e.g., color), arguments of functions (e.g., multiplier and multiplicand of a multiplication), and “cases” of verbs in sentences (e.g., “agent”). Any generalized attribute of this sort has two important pieces: (1) the particular entity that becomes the value for the attribute in an instance of the Concept, and (2) the functional role which that entity fills in the conceptual complex. A Role is a formal entity that captures both of these aspects in a structured way, by packaging up information about both the role filler and the functional role itself.

In KLONE, the substructure of a Role indicates the following: the type of entity that can fill the functional role, how many fillers of that role are to be expected, whether it is important in some way to have the role filled in an instance, and the name of the functional role itself. Notice, then, that the formal entity, Role, is somewhat more than a description of either the potential filler or the functional role alone. It is a very special type of epistemological entity that ties together the functional role, the context in which that role is played, and the (set of) filler(s) of the role. Figure 16 schematically illustrates the internal structure of a Role and its place in a Concept (Roles will henceforth be pictured as small squares, while Concepts will be depicted as ovals).

As just mentioned, while the “internal” Role structure indicates information about the particular fillers in themselves, the Role itself is the meeting place for information about how those fillers fit into the entire conceptual complex. It is the Concept's set of Structural Descriptions (SDs) that is the source of information about how role fillers interact with each other. Each SD is a set of relationships between two or more of the Roles. Just as a Role indicates that for any instance of the Concept there will be the appropriate number of fillers (with the corresponding characteristics) for the given functional role, an SD indicates that any instance of the Concept will exhibit the relationships specified in that SD. So, for example,
the Concept for a simple arch, which has three bricks as its parts, might be factored as in Fig. 17 (SDs are indicated by diamonds).

In this figure, Role R1 expresses the fact that this kind of arch has one LINTEL, which must be a WEDGE-BRICK. The RoleD link expresses the relationship between the Concept ARCH and one of its Role descriptions; the V/R (Value Restriction) link points to the type predicate that must be true of the eventual role filler; the Number link indicates the restriction on the number of fillers of the role; the Modality link indicates the importance of the attribute to the Concept; and finally, the RoleName link names the relationship (conceptual, not epistemological) between the filler and the whole. R2 similarly indicates that any example of this type of object has two UPRIGHTs, which are BRICKs. R3 defines the generalized attribute of VERTICAL-CLEARANCE. The Modality INHERENT means that, while every arch has one of these, knowing its value is not critical to the recognition of some object as an ARCH. In addition, DERIVABLE means that the value can be computed from the values of the other Roles, once they are found. As for the SDs, S1 is a set of relationships that expresses how every UPRIGHT supports the LINTEL, S2 specifies that no two UPRIGHTs touch each other, and S3 embodies the definition of the VERTICAL-CLEARANCE in terms of the LINTEL and an Individual Concept, GROUND. We now turn briefly to the internal structure of these relational parts of the Concept.

Let us say that we want to define the VERTICAL-CLEARANCE of an ARCH to be the distance between its lintel and the ground. There will thus be some Concept related to DISTANCE in one of the SDs of ARCH.

To determine the exact nature of this Concept, let us look at the way that we have expressed the relationship in English: first, the definite determiner for "ground" indicates that we mean a unique individual. To reflect this, our network would have an Individual Concept, GROUND, which corresponded to that "constant." Further, there should be some individuator of the DISTANCE Concept with one of its Roles satisfied by GROUND. In Fig. 18, we illustrate this partial state of affairs—D1 is an individuator of DISTANCE (indicated by the Individuates link), and the fact that its TO Role (R1) is satisfied is captured by R2, whose Satisfies link indicates the appropriate Role, and whose Value link points to the filler. Now—back to our English description—we have still to account for "its lintel" and "the distance." By "its," we mean that for each arch, there is one lintel, and that is precisely what we meant by the Role R1 in Fig. 17. The "the" with "distance" then follows as saying that for each instance of ARCH, there is one unique distance involving the lintel of that arch. Thus, what we thought was an individuator of DISTANCE, is not quite—it has Role fillers tied down to lintels, but not to a single constant one.

The fact that the FROM Role of D1 is to be "filled" not by a constant, but by a type of existential, makes it a different sort of entity than R2. It is not quite a general Role, since it can only be filled by the lintel of a particular arch; nor is it a filled Role. Instead, it is an argument of a Concept that is parameterized by another Concept—D1, parameterized by ARCH. Once a particular arch is selected, the filler of the corresponding DISTANCE's FROM Role is fixed. We call this type of Concept a "Parametric Individ-
ual” (ParaIndividual), and express it as in Fig. 19. In this figure, the double oval represents the ParaIndividual, which is linked to DISTANCE by a ParaIndividualates link. The double square is a “Coref Role,” which equates (as coreferential) the filler of the FROM role of the particular distance in some instance of ARCH with the filler of the LINTEL role for that same arch. CorefValue links the equated Roles, and CorefSatisfies performs an analogous task to that of Satisfies in an ordinary filled Role.

3.1.1. Epistemological Relations for Structuring Concepts

Our notions of Concept, Role, and SD give us the picture of structured conceptual objects schematically illustrated in Fig. 20. This structure implies that a knowledge representation language that is based on structured conceptual objects must account for at least the following relationships:

1. the relationship between a Concept and one of its Roles,
2. the relationship between a Concept and one of its SDs,
3. the “internal” structure of a Role—the relationship between a Role and one of its facets,
4. the “internal” structure of an SD,
5. relationships between parts of SDs and Roles.

In SI-Nets, we account for these explicitly as link types, most of which were illustrated in the above figures. Thus, the primitives in this notation are epistemological (knowledge-structuring) relationships that compose formal representational objects. It should be clear at this point that there is no sense to having links like COLOR and ASSAILANT in the same formalism with links for epistemological operations (nor links like AND and PRED, for that matter). The relationship between a Concept and a Role/Filler structure is not the same as that between the object that the Concept represents and the thing that fills the functional role for that object.

The semantics of each of these links is, of course, built into the interpreting functions that operate over the network structure. While I shall not detail that interpreter here (except briefly; see Section 3.4), it should be noted that with a small, predetermined set of link types, a fixed interpreter can, at least in principle, be designed. In languages that claim to have no primitives at all, the status of an interpreter is in question. *

3.2. Epistemological Primitives for Inheritance

One type of epistemological relation that we have so far glossed over is that which connects formal objects of the same type—Concept to Concept, Role to Role, and SD to SD. This type of link is a critical one in the SI-Net scheme, since it accounts for inheritance. For example, as mentioned, individuation is a relationship between Concepts, such that there is always some description (Concept) that is being individuated. That Concept is composed of various subdescriptions, all of which must be satisfied by the individuating Concept. Not only is there a relation between the two Concepts involved (i.e., Individuates), there is a set of subrelations between the generalized attribute descriptions (Roles) of the parent Concept and the values of those attributes in the individuator (i.e., the relation Satisfies expresses this in the above examples).

* See Smith [1978] for a philosophical account of the place and nature of interpreters in knowledge representation schemes.
The notion of inheritance is broader, however, than just the definition of an Individual Concept by a more general Generic Concept. "Subconcepts," themselves also Generic, can be formed from Generic Concepts by restricting some of the subparts of the description embodied by the Generic Concept. As we have seen in the history of semantic nets the formation of more and more specific descriptions is an important common feature, and taxonomic hierarchies depend on this for the backbone of their structure. There has generally been a single link (e.g., IS-A) to specify inheritance along sub/superconcept chains, and the assumption has been that everything relevant to a general class (e.g., MAMMAL) is relevant to its more specific subclasses (e.g., DOG, CAT). Looking at this with our epistemological eye, however, we find this to be an oversimplification of a multifaceted relationship.

The Roles and SDs of a parent Concept each contribute to the inheritance of a subConcept. Thus the inheritance link is effectively a "cable" carrying down each of these to the inheritor; the Roles and SDs must be transmitted as a group, since they do not have an existence independent of the Concept of which they are parts. Just as Fahlman's "virtual copy" link implies that all parts of the structure are immediately available at a subconcept, we think of inheritance as a structured epistemological relationship between Concepts.

Further, properties are usually not all inherited intact, but instead are often modified so as to give the subConcept a more restricted definition than the parent Concept. In that case, each of the modifications must be represented in an explicit and precise way. Figure 21 sketches the set of epistemological relations between a parent Concept and one of its descendants. For each Role and SD that is to be modified in some way, we must say precisely what type of modification applies, and what Role or SD the modification applies to. The latter is indicated by an inter-Role or inter-SD link stating the relationship between the original Role or SD and the new, modified one. The modification itself is then indicated just as if the modifying Role were a new Role description. KLONE currently allows three types of Role modification (satisfaction, or filling; differentiation, or the creation of subRoles; and restriction of the Role constraint). At the moment, only one type of SD modification (preempting) is provided. These relationships are explicitly indicated by appropriate links with unambiguous interpretations.

The reader should consult Brachman [1978a] for further details on KLONE. Here I have attempted only to illustrate the flavor of relationship for which it is necessary to account. More specifically, it is the job of an epistemological level formalism to provide internal Concept-, Role-, and SD-structuring relationships, and inheritance-specifying inter-Concept, inter-Role, and inter-SD relationships.

3.3. The "Conceptual Coat Rack"

In many of today's representation languages, there is a way for the knowledge base designer to go directly to the language in which the system is implemented (e.g., LISP) in order to express certain facts or associate certain procedures with network structures. Such "escape" mechanisms are used either when the knowledge to be expressed is too complex to be represented in the network itself, when knowledge about the network itself is to be encoded, or when certain procedures are to be triggered by operations on the data base. With the work of Smith [1978], the epistemological import of "procedural attachment" is now clear. There are, according to Smith, two different types of attachment that are most often confused under the guise of "procedural attachment": (1) "metadescription," wherein knowledge about knowledge is expressed in the same network language as the primary knowledge; and (2) interpretive intervention, in which direct instructions to the interpreter are expressed in the language that implements the interpreter itself.

In the case of metadescription, the interpreter is being asked to make a type or level jump when processing a Concept. Metainformation is information about a Concept (or Role or SD) as a formal entity, and is not information about the thing(s) that the Concept describes. To support this kind of information, KLONE provides an explicit link to a node representing a separate sense of the Concept as a formal entity. This link is called a "metahook," and it can attach to a Concept, a Role, or an SD. Metahooks always point to Individual Concepts, and those Individual Concepts express knowledge in the normal KLONE way—except that their "references" are formal entities in the net, and not objects in the world.

KLONE provides another kind of hook, the "interpretive hook" (ihook), for attaching interpreter code directly to a Concept, Role, or SD. The code

---

* In addition, further properties can be added to form more specialized Concepts, e.g., PRIME-NUMBER from NUMBER.

* This is the case in the Levesque and Mylopoulos "instance hierarchy," for example.
pointed to by an ihook must be constructed from interpreter primitives (e.g., functions like “CreateConcept,” “SatisfyRole”), and the ihook must specify the place in the interpreter from which the code is to be invoked. These hooks are not intended as escapes in which arbitrary information can be encoded when the formalism makes it hard to express a fact about the world, but as a means of direct advice to the interpreter with clear import.

The two kinds of hooks express important relationships between the KLONE interpreter and data base. These relationships are different in nature from those expressed by the intentional, structure-building links discussed above. They allow us to look at the part of the network built out of those links as a structure on which to hang knowledge about knowledge or advice to the interpreter—as kind of a “conceptual coat rack.” The knowledge-structuring relationships can be thought of as forming a representation “plane” out of which hooks emerge orthogonally.

3.4. Interpreting KLONE Structures

While I have given the impression that there is a single KLONE interpreter that deals with the node- and link-types described, that is a bit misleading. KLONE is implemented in INTERLISP as a set of interpreter primitives, all of which together, in some sense, form an “interpreter.” However, these primitive functions for building, accessing, and removing structure are not organized into a single cohesive program. Instead, they may be used in combination by higher-level functions (matching, reasoning by analogy, deduction, etc.) to construct and maintain a KLONE data base. Each function guarantees structural integrity, and the set of functions together constitute the only possible access to the KLONE structures. In this way, Concepts, Roles, and SDs are like abstract data types in CLU [Liskov and Zilles, 1974]. The functional interface provides a clean, implementation-independent definition for the types of entities that KLONE supports.

The principal motivation for providing a set of primitive functional pieces out of which “higher-level” procedures can be built, and not a particular set of matching, deduction, etc., procedures, is that it is felt that we do not have a clear enough understanding of these issues to allow us to provide powerful procedures at this higher level. Experience with matchers in the field in general has been equivocal, and we have chosen instead to provide a basic set of tools for building different variants on an experimental basis. Since there is no general understanding of things like matching and reasoning by analogy, it seems wise not to commit the basic package to some ad hoc set of processing routines. This does not mean to say, however, that there do not exist such higher-level routines for KLONE—we have, in fact, been experimenting with a variety of approaches to structure-matching, paraphrasing, question-answering, and situation recognition. KLONE is well-suited to some of these tasks, and where possible, we have provided the obvious functions. With some of these, we are investigating the use of “parallel” marker-passing algorithms (see Woods and Brachman [1978], for example).

The KLONE functions depend on the fact that the set of connections between Concepts, Roles, and SDs is fixed in advance. In order to implement, say, a function that finds a (possibly inherited) facet of a Role, we need to be able to anticipate all possible forms of inheritance that will be encountered. The function can then look for immediately accessible values, and if not found, can call a variant of itself recursively, depending on the type of Role inheritance connector it encounters. A complete set of KLONE inheritance functions, including the provision for multiple super-Concepts and multiple super-Roles, has been implemented, based on the small set of possible inter-Role relationships.

Since the user of KLONE “sees” only abstract structures for Concepts, etc., it is not necessary to think of the network as a set of nodes interconnected by links, but instead to view Concepts as sets of Roles and SDs, etc. The functions deal only with those entities (and their “epistemological” relations), and never attempt to make or break simple local linklike connections. This is important, considering that structured inheritance is a central feature of KLONE; a “cable” contains many connections that are not independent. One problem with the traditional semantic network metaphor in general is the apparent independence of each link from all other links.

There are currently two significant uses being made of the KLONE interpreter package. One involves a natural language understanding system that combines general English ATN-based parsing with the benefits of “semantic grammar” [Brown and Burton, 1975; Burton, 1976]. A KLONE taxonomy has been built that encodes semantic categories for certain types of phrases, and the parser, guided by a very general grammar of English, interacts with this taxonomy to build up the representation of a sentence.

The Concept–Role paradigm is ideal for expressing the relationships between categories like “person-NP” and its possible modifiers, since it provides a completely general case-definition facility. Further, the interpretations of sentences are built incrementally from their syntactic representations using the ihook and metadescription facilities to map syntactic structures into those of a conceptual network. The conceptual net expresses the relationships between the entities discussed in the sentences, which at the moment, include people, places, and research topics. Structural Descriptions play a large part in handling paraphrase utterances and determining answers to queries.
The other domain to which the package has been applied is the description of general graphics knowledge and how to use the display facilities of a bit-map terminal. This knowledge includes coordinate system transformations, projections of entities onto display surfaces, and interrelations between actual domain objects (like ships and land masses) and their corresponding display forms. For example, one might incorporate into the general knowledge base the desire to see ships displayed as circles with centers at their projected positions, and augment that with the instruction to display ships with radar with a special symbol. Once particular ships were described to the system (and incorporated into the portion of the network dealing with domain objects), their displays would be hand-ed automatically. The knowledge is encoded in KLONE so that it will not only be useful in producing the displays, but will also be available for discussion and easy manipulation by the system's user.

CONCLUSIONS

In this chapter, I have examined the history of the "semantic" network, looking for the major conceptual breakthroughs that have made it such a popular representation technique. It was found that through that ten-year history, at least five different interpretations of nodes and links have crept together to create confusing languages with limited expressive power. In the last two years, efforts have been mounted to crack that expressive deadlock; these efforts have concentrated on the logical status of network primitives, and have begun to take a hard look at the foundations of network representations. At the same time, the field has begun to see higher-level structures (e.g., partitions, frames) imposed on nodes and links. These structures appear to be useful and significant, but no comprehensive effort has been made to understand exactly what their status is.

In Section 2, I postulated a set of conceptual levels for interpreting primitives in semantic networks. The four that were immediately apparent from the history of the semantic net were the implementational, logical, conceptual, and linguistic levels. Each of these has had at least one (perhaps implicit) proponent in the literature. In addition, to account for some more recent aspects of knowledge representation, and the standard descriptive use of network concepts, I proposed an intermediate level to account for the internal structures of Concepts, and the relations of inheritance that exist between them. I called this the epistemological level of knowledge representation.

Section 3 attempted to make more apparent the kinds of relationships that an epistemological level representation should express. It was noted that descriptions of functional roles in complex objects and descriptions of the fillers of those roles were a critical part of knowledge about the world, and that, in addition, the meaning of a functional role was bound up in a set of relationships between its fillers and the fillers of other roles in the object. Given this interpretation of structured objects, the set of epistemological relations that a formalism must account for becomes clear. Finally, I tried to illustrate how a network language might account for all of these relations with explicit epistemological links, and how the structure thus formed could be used to hang information for the network interpreter.

In conclusion, it is in general useful to try to produce a knowledge representation language that is built on a small, fixed set of primitive node and link types. Settling on a fixed set of primitives, with well-understood import in terms of the operations of a particular level, enables the network designer to construct a well-defined and fixed interpreter. In addition, consistency at a single level affords the best position from which to achieve adequacy toward the level above.

ACKNOWLEDGEMENTS

I would like to thank Norton Greenfield, Austin Henderson, Rusty Bobrow, Bill Woods, and especially Martin Yonke, for their great help in understanding and implementing KLONE.

REFERENCES


