AD-A060 332   BOLT BERANEK AND NEWMAN INC CAMBRIDGE MASS   F/G 5/7
THEORETICAL STUDIES IN NATURAL LANGUAGE UNDERSTANDING (LANGUAGE--ETC(U)
SEP 78   R J BRACHMAN, W A WOODS   N00014-77-C-0371
UNCLASSIFIED   BBN-3888   NL

1 OF 1
ADA
060332

END
DATE
FILMED
12-78
DDC

Bolt Beranek and Newman Inc.

LEVEL II

(12)

Report No. 3888

Theoretical Studies in Natural Language Understanding

Annual Report, 1 May 1977 to 30 April 1978

September 1978

Prepared for:
The Office of Naval Research

Unclassified.

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER BBN Report No. 3888	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER

4. TITLE (*and Subtitle*) THEORETICAL STUDIES IN NATURAL LANGUAGE UNDERSTANDING Annual Report 1 May 1977 - 30 April 1978	5. TYPE OF REPORT & PERIOD COVERED Annual Report 5/1/77 - 4/30/78 6. PERFORMING ORG. REPORT NUMBER BBN Report No. 3888
7. AUTHOR(s) Ronald J. Brachman	8. CONTRACT OR GRANT NUMBER(s) N00014-77-C-0371
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, MA 02138	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Information Systems Arlington, VA 22217	12. REPORT DATE September 1978 13. NUMBER OF PAGES 73
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)	15. SECURITY CLASS. (*of this report*) Unclassified. 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Epistemological representation, KLONE, knowledge representation, natural language understanding, semantic networks, structured inheritance networks.

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This report describes progress made during the first year of an ongoing project, the goal of which is the creation of an effective representational framework for dealing with natural language information. By "natural language information," we mean data that ranges rather freely within the scope of natural conceptual structures, and does not fall into any simple paradigm. The achievement of such a framework requires

Unclassified.

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

20. Abstract (cont'd.)

a synthesis and development of fundamental ideas in representation, semantics, epistemology, and computer algorithms and data structures. This report describes the state of our understanding of the basic epistemological issues at the end of the first year of the project, and the status at that point of a formal representational system (KLONE) for natural language information. Although the framework is still under active development, it currently demonstrates strong capabilities for representing the inheritance of component parts of complex structured concepts, and provides a continuing stimulus for ongoing research.

LEVEL II

⑥

THEORETICAL STUDIES IN NATURAL LANGUAGE UNDERSTANDING

(Language Understanding).

⑨ Annual Technical Progress Rep[redacted]t.

1 May [redacted]77 - 30 Apr[redacted]78,

Program Code No. ID 121408	Short Title of Work: Language Understanding
Contract No. ⑮ N00014-77-C-0371	
Name of Contractor: Bolt Beranek and Newman Inc.	Contract Expiration Date: ~~30 April 1979~~
Effective Date of Contract: 1 May 1977	⑩ Ronald J. Brachman Principal ~~Investigator:~~ Dr. William A. Woods ~~(617) 491-1850, ext. 361~~
Amount of Contract: $180,000	Scientific Officer: Marvin Denicoff

⑪ Sep 78

⑫ 81p.

ACCESSION for	
NTIS	White Section ☒
DDC	Buff Section ☐
UNANNOUNCED	☐
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

Sponsored by
The Office of Naval Research

⑭ BBN-3888

This research was supported by the Office of Naval Research under Contract No. N00014-77-C-0371.

DDC
RECEIVED
OCT 23 1978
D

060 100

Gee

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

Theoretical Studies In Natural Language Understanding

Annual Report

Foreword

This report describes progress made during the first year of an ongoing project, the goal of which is the creation of an effective representational framework for dealing with natural language information. By "natural language information", we mean data that ranges rather freely within the scope of natural conceptual structures, and does not fall into any simple paradigm. Such data is generally not encoded in predetermined formats (and cannot be fit into a small number of rigidly formatted templates), and cannot be understood by merely decoding some input string. This type of information is characterized by utterances that make use of complex natural language phenomena, such as ellipsis, pronominal reference, relative clauses, natural quantification, adjectival and adverbial modification, and attention focusing transformations.

Our intent is to create general, powerful representations of knowledge so that a machine might use basic, "common sense" concepts to understand natural language utterances. Understanding would be reflected in the ability to assimilate new information discussed in terms of existing concepts, to answer questions about a current data base, and to perform actions where relevant. In addition to overtly visible actions made in response to natural language inputs, inference may often be necessary to make up for information not explicitly represented in the machine's knowledge base. The framework that we develop to represent explicit knowledge about the world should support in a fundamental way the drawing of conclusions not explicitly stated in a text. Besides an operational inferential capability, this demands a generous a priori understanding of basic information that itself is never explicitly present in natural dialogue.

Therefore, in addition to our basic concern with abstract representational and inferential adequacy, we have focused much of our effort on the representation of basic, common sense information about people, places, physical objects, and mental abstractions. In particular, we have concentrated on knowledge about papers, drafts, research topics, research institutions, employment, conferences, etc.

In order to even begin the construction of concepts for people, places, papers, etc., we have found the need to spend much of our effort understanding the prerequisites to knowledge representation. That is, this year we have concentrated on laying an adequate epistemological foundation on which to build natural language conceptual knowledge. First, we attempted to gain some insight into the history of associative formalisms for representing conceptual information. This led to an appreciation of a set of important differences among the representational primitives of various notations (the "semantic networks", in particular), and consequently to a characterization of five "levels" of primitives for representing knowledge. One of these, the "epistemological level", is becoming an important focus of representation research, and we have taken to be our task the creation, use, and understanding of a set of epistemological primitives out of which we might construct effective representations of conceptual knowledge. The inspiration for this foundational work, which constituted the major part of this year's effort, was research reported in [Brachman, 1978b]. We have taken the representational formalism first developed there -- the "Structured Inheritance Network" -- and expanded its account of structured conceptual objects and inheritance. In addition, we have begun the implementation of an experimental system (called "KLONE") that will allow us to embody and test our representational conventions and inference algorithms. We are currently integrating the representational system with a general-purpose ATN-based parsing system, and we are preparing to handle various types of anaphoric reference as well as natural language quantifiers. We are also engaged in developing inference algorithms for our network structures. Since the work on the natural language interface and the algorithms has just begun in earnest, it will not be reported on here.

Epistemological Primitives for Factoring Knowledge

Ronald J. Brachman

Perhaps the most difficult task facing us when we sit down to try to represent* some segment of our knowledge is our very first: at what level of abstraction do we begin to break this knowledge down? When designing data structures to be manipulated by a single-minded program, we find it easy to determine the "representational grain". But when trying to capture the details of our knowledge about a particular domain of expertise, in order to support a general cognitive system whose goals in manipulating the structure we cannot determine in advance, we are at a loss to determine the conceptual size of the units of our knowledge. For example, if I want to express some general knowledge of people, research institutions, and research topics in, say, predicate calculus, how am I to determine the most useful set of predicates? Is the predicate, PERSON(X), sufficient? Do I need ARM(X), and LEG(X); how about FINGER(X) etc.? How detailed do I make the primitive predicates so that I support all distinctions that my reasoning system will need to make, yet avoid a proliferation of

---

* There is an even more fundamental difficulty here: how well do we understand what it would mean to have something "represented"? For this discussion I will hold this issue in abeyance, although it is critical to understand before the enterprise can be measured as successful.

useless (and perhaps ontologically meaningless) epistemological minutiae?

Besides problems with the grain size of a representation, one very quickly encounters problems with basic types of conceptual objects (especially in logical formalisms like predicate calculus) - do I make my conceptual objects noun-like or adjective-like? For example, does conceptualizing the predicate, PERSON(X), force me to think of "person-ness" being attributive rather than of persons as objects? And when and where does this make a difference?

This is the type of question facing the "knowledge engineer". The intent of this paper is to examine these questions and attempt to offer some help toward their resolution.

1. Implicit Decisions in Representation Languages

Where would the hapless representer-to-be look for help? The obvious place is to the representation language that is being considered for the encoding of the data. Each such language provides for its user (the knowledge engineer) a set of object types and syntactic conventions, which together suggest how to factor concepts of the domain. That is, by adhering to the representational conventions of a language, we have certain decisions made for us - the language embodies the set of distinctions that its designer has deemed meritorious, by pre-specifying the types of entities and relationships that can be encoded and the level of detail of structure that can be expressed. The primitives of a language implicitly embody the **epistemology** that the language's author believes is the way to look at the conceptual world.

To a large extent, the last ten years' representation languages have had their epistemologies all too implicit. In particular, "semantic networks" have achieved great popularity in systems that deal with natural language information, yet only very recently has the **adequacy** of these representations come into question [Woods, 1975; Brachman, 1977, 1978b]. Each version of semantic net had implicit in it a set of knowledge-factoring principles that were given little explicit acknowledgment. In some cases, it was predetermined that two verbs could differ in the set of "cases" they took [see the work of Simmons et al.], but this failed to account for the differences between verbs with identical case frames. In other cases, logical predicates and connectives were deemed the base level of representation [Schubert, 1976; Cercone, 1975], while still others allowed both logical constructs and cases [Hendrix, 1975a, b, 1976]. And still others expressed the belief that very high level primitives - those of a natural language itself - were the only kind sufficiently vague to allow us to properly factor knowledge (OWL - see [Martin, 1977; Szolovits, Hawkinson & Martin, 1977]. At least in the case of logical networks, formal adequacy was an understandable and achievable goal, and eventually did become an explicit target for representation system designers. But in most cases, it was not clear what the representational theories were that underlay the semantic net data structures.

One of the goals of this paper is to help sort out the epistemological morass of the history of semantic nets and kindred structures. In order to be of some help to the frustrated knowledge engineer, who wishes to find the appropriate level of representation and type of conceptual object, we should first try to assess the implicit views that have found their way into various extant representational formalisms. Therefore, I will first spend some time looking at the nature of the **primitives** of network-related representation languages, before examining the nature of an explicit epistemology representation system.

2. Representational Primitives in Semantic Nets

Semantic nets were developed primarily to support **associations** between "word concepts" used to express the meanings of English words [Quillian, 1966,67,68,69]. While the conception of word meanings has changed, and nets have been broadened to cover more general conceptual information, the associativeness has remained the trademark of this class of formalisms. As a result, the primary emphasis has been on the meanings of the **links** in the network [see Woods, 1975]. Only more recently [Hendrix, 1975a, b, 1976; Brachman, 1977, 1978b; Levesque & Mylopoulos, 1978] has the focus shifted to the types and structure of **nodes** in these languages. Node structure as well as link meaning has important consequences for the representer using the network system.

In order to help understand the implications of the particular sets of links and node-types offered as "built-in" by semantic net schemes, I will briefly present a set of five abstract **levels** of representational primitives. The level that characterizes a representation scheme (independent of the adequacy of the particular scheme) has a strong impact on the way knowledge will be factored in order to be expressed in that language, since its primitives represent the finest distinctions that the language is capable of expressing. These levels range from the mechanistic **implementation** level through the intermediate **logical**, **epistemological**, and **conceptual** levels, to the most information-laden **language** level. It should be pointed out that the "primitives" I am referring to here are representational, or constructional primitives. While each language necessarily provides a repertoire of primitive syntactic elements for constructing structures, many systems also offer a set of particular concepts (e.g., "THING", "PHYSICAL-OBJECT", "NOTHING", etc.) as built-in "informational" primitives. The latter, while of interest and importance in their own right, are not of concern to the discussion that follows.*

The kinds of associations that semantic net languages support range from nondescript links which act essentially as pointers, all the way to those of arbitrarily complex conceptual relations. Many (if not all) of these languages provide more than one kind of primitive, and it is not the intent here to characterize the languages as purely exemplary of any level, but instead to help clarify the interaction between representational entities. In fact, it will generally be stated that a language at a given level can always be implemented in terms of the primitives of the level below.

This claim, however, is subject to the representational integrity of the "higher"-level language. If I have a primitive structured nominal concept, say, that I claim in my language to directly represent some class of entities, R, in the world, then supposedly the actual representation of R cannot be broken down in terms of lower-level primitives. Yet a lower-level primitive Predicate, say, R(x), claimed to represent "R-ness" in the world,

---

* See [Winograd, 1978] for a discussion of other dimensions along which primitives may be judged.

might look as if what it represented was the same as what the higher-level primitive represented. However, there is something subtly different between the Concept of R and the Predicate of R-ness, having to do with what is claimed to exist outside the knowledge base. One claims "objectness", while the other claims "attributeness", and it is problematic to consider "implementing" the object in terms of the lower-level attributes. If this ultimately turns out to be a critical difference (and our feeling is that it will), then it belies McDermott's [1978] contention that this is only a matter of "notational engineering". It is, in fact, a true matter of **representation.**

2.1 The implementational view

The first view of semantic nets that we will consider is the one that takes them to be associational data structures, with no representational import. The constructional primitives of such schemes are pointers and atoms, and can be used in any way the knowledge engineer desires. No semantics are ever claimed for the elements of this lowest level - the most significant claim that is made is usually about the **indexing** facility provided by the associational links.

Very few systems have been proposed that can be considered purely at this level (the earlier work of Shapiro [1971a] is perhaps the best example), but many systems are littered with elements that are implementational. For example, in most systems with an "escape" to the interpreter's language (e.g., LISP), the escape mechanism and LISP code are not explainable with the same type of semantics that applies to the rest of the language. As another example, **partitions** in their pure form [Hendrix, 1975a,b] are simply implementational conveniences for grouping nodes, without representational significance. They can be ascribed such import by elevating them to the level of the system as a whole (as, for example, with "focusing contexts" [Grosz, 1977], and scoping of logical quantifiers [Fikes & Hendrix, 1977; Hendrix, 1978]), but the basic partitioned network system offers them as a non-semantically supported mechanism. Another view of networks as merely implementational is expressed in [Nash-Webber & Reiter, 1977], wherein they are seen as labeled graphs which simply implement a logical form.

An interesting sidelight here is that the recent push to "frames" is almost strictly an implementational one (see [Roberts & Goldstein, 1977]). Frames are made of "slots", which are simple structural subparts without any representational connotations. Frames become fancy data structures, and not representational structures.

2.2 The logical level

The first jump to primitives with some representational import is usually made at the level of **logical** elements. The semantics of constants, predicates, and propositions is relied on for networks that express the same structures as can be expressed in linear predicate calculus (PC) forms. When primitives in representation languages take on this logical form, the language becomes an alternative surface form for predicate calculus (parentheses and adjacency are replaced by pointers) - usually with some added "indexing" facility.

The reasons for which one designs a representation language with logical primitives are obvious and sound: predicate calculus is a well-understood and well-founded system for manipulating formal symbols, and has an apparently adequate semantics.* Networks imposed over predicates and propositions can provide an easy way to get from an assertion to other "associated" assertions, and the basic inheritance mechanisms provided by networks can be used to abbreviate the expression of complex statements. The principal proponents of logical networks (Schubert [1976], Cercone

---

* The semantics normally ascribed to these notations is Tarskian truth-theoretic semantics. But, as Reiter [1978] points out, there are subtle ways in which default assumptions for networks have made them different logics than in PC, which are perhaps more appropriate for certain types of "common sense" reasoning.

[1975, Cercone and Schubert, 1975], Hendrix [1975a, b, 1976, 1978; Fikes & Hendrix, 1977] and Shapiro [1975, 1977, 1978]) generally argue that their networks provide real advantages over predicate calculus representations, including that they "seem more natural and understandable" [Schubert, 1976, p. 164]. It seems clear, however, that with the right "closed world default" assumption [Reiter, 1978], the formal expressive power is no different in either case, although certain inference algorithms may appear more easily facilitated by one or the other representation, and the subject is one of some current debate [see Bobrow, 1977]. In any case, the logical level is the one level of network representation that comes into direct contact with classical notions of semantics, and the only level wherein claims of adequacy are not made solely on empirical or intuitive grounds.

2.3 Epistemological structuring

One concern of advocates of languages at "higher" levels is that the primitive elements of logical languages are too "small" to be easily compounded into meaningfully large representation structures. While PC is logically adequate, there is reason to believe that knowledge representation can be done with more communicative power by resorting to a set of primitives that (in the right context) are not logically primitive. This is the substance of primitives which I shall call **"epistemological"**.

The focus at the epistemological level shifts away from predication, and onto "objecthood." Several languages are under development in which the **structured conceptual object** is the principal representation device, and where the primitives are structure-types and relations that are used to bind together such objects (see [Brachman, 1977, 1978a, b], [Bobrow & Winograd, 1977], [Roberts & Goldstein, 1977], [Fahlman, 1977], [Sridharan, 1978], [Levesque, 1977; Levesque & Mylopoulos, 1978], [Ph Hayes, 1977a,b], [Hays, 1973]). There has also been a marked effect of this thinking on languages whose primary level is that of logic: certain uses of partitions (notably for "delineations" - see [Hendrix, 1975a, 1978]) make use of logical primitives to create structured definitions for concepts, and Cercone, et al. [Goebel & Cercone, 1978] have resorted to non-logical notions like "topic" to elevate some aspects of their logical nets to the epistemological level.

The difference between logical level and epistemological level paradigms is subtle, and while it at first might appear to be an arbitrary one, I believe it to have some significance in the shape of the next generation of representation languages. The primary difference is the shift from an orientation toward attribution to a concern for the internal structure of perceived objects. The focus on objects leads to investigation of the functional **roles** to be played within the object (often referred to by their implementation-level counterpart's name, "slots), and the kinds of structured inheritance relations that can exist between objects. Details on the concerns of epistemological languages are discussed in [Brachman, 1978a], and are touched on in the section describing KLONE that follows this one.

2.4 Conceptual level concepts

While the concern for concept-structuring epistemological primitives is a recent development, one of the oldest trends in semantic nets is that of providing primitive concepts and relationships. The principal influx of this kind of structuring tool came at the time of incorporation of linguistic case structures ([Simmons, Burger, & Schwarcz, 1968; Simmons & Bruce, 1971], [Norman, 1972]). Perhaps the strongest influence on work at this conceptual level was the work of Schank [1972, 1973a,b; Schank & Rieger, 1974], in which a small set of primitive ACTs and case relations was proposed. In a system based on **conceptual dependency**, it is claimed, any concept can be broken down into a canonical complex of the primitive actions and cases, inferences can then be drawn from the primitives, and paraphrases can be generated. This paradigm has worked well in the domains to which it has been applied, and is the only one (to my knowledge) to have stood by a fixed set of conceptual primitives. Further, the fact that inferences are based only on a predetermined set of entities is appealing in its cleanliness.

For pure conceptual level formations, the central question is what particular concepts and cases are to be taken as primitive. An epistemological formalism, on the other hand, provides the facility for defining an open-ended set of basic concepts and cases, thereby implicitly assuming that while the notion of "case" (role) is primitive, no particular case necessarily is. These two views have different psychological and theoretical implications, some of which might be empirically testable.

2.5 Language level formalisms

One of the salient features of conceptual level networks is their language-independence. This reflects the predilection of the designer of such a language that "it should reflect the important properties of relationships inherent in the world rather than those in the constructor of the KR's native language" [Schank, 1977, p. 988].* We might, however, consider an alternative approach, in

---

* One might in fact contrast this with the view that the epistemological system constructor wants to account for more than just the world; that is, what we might call "epistemologically possible (thinkable) worlds." It would follow, then, that logical networks would account for "logically possible worlds."

which cognition is not possible without language. This commitment - "to 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] - is the founding principle for OWL [Martin, 1977; Szolovits, Hawkinson & Martin, 1977].

In OWL (and presumably other formalisms like it), there is a small set of structuring relationships that tie together language-specific concepts. These relations are small concessions to the need for a syntax for the language, and are a bit problematic to analyze. They seem to be either epistemological, in that they in some sense make claims about the underlying structure of all concepts,* or implementational, in that in some cases "the meaning of that connection [i.e., the "attachment" of B to A] will depend completely on what A and B are and on whatever is interpreting the connection" [Szolovits, Hawkinson, & Martin, 1977, p. 9]. Everything in an OWL network is essentially "primitive" with respect to any new piece of information. Once new information is added, the primitive set is changed. The sense of everything depending on what has preceded pervades the OWL network structures. As the OWL authors claim, this highest level provides "almost

---

* To be more specific, OWL claims that concepts are always derivable from more general concepts by a simple, noncommittal "specialization."

trivially" the expressive power of English in a computer language - at the expense of this feeling of "it all depends" and possibly begging the question of the language's semantics.

2.6 A note on the history of semantic nets

The history of semantic network formalisms can be analyzed in an interesting way by viewing network primitives at the various levels just introduced (see [Brachman, 1978a] for a detailed survey of this history). The most prevalent early forms (e.g., [Carbonell, 1970a,b], [Winston, 1970], [Simmons, 1973]) involved conceptual level links, in what were attempts to abstract away the language part of natural language information. Thus the conceptual universals were the substance of each network's epistemology, and debate tended to be over the number and nature of "conceptual cases." Another early thread was the desire to establish extremely flexible tools that network language designers could use to build their own languages. Shapiro [1971a], for example, called attention to the distinction between conceptual ("item") relations and lower-level "system" relations, and showed how the higher-level ones could be built on top of an implementation level set of primitives. Further, one can trace the influx of logic into semantic nets, as the frustration with the lack of constraint in implementation systems, and the lack of semantics in conceptual ones, eventually led Hendrix [1975a,b], Shapiro [1975], Schubert [1976] and others into primitives taken more or less directly from predicate logic. With mechanisms for expressing predicates, propositions, and explicit quantification, the logical adequacy of the networks was easily determined. At that point, the issues centered around the advantages of various particular representations for quantifiers, the adequacy of particular schemes, and the purportedly questionable need for any higher level constructs.

Recently, the communicative clarity of explicitly quantified propositions, etc., has come into doubt. Even the hard-core logical level designers have attempted to augment their networks with "frame"-like constructs to facilitate ease of concept expression and reasoning (see [Goebel & Cercone, 1978], for example). Several representation language efforts ([Bobrow & Winograd, 1977], [Brachman, 1978a,b], [Fahlman, 1977], [Levesque, 1977], [Ph Hayes, 1977a,b], [Sridharan, 1978], [Srinivasan, 1976], and [Roberts & Goldstein, 1977]) have proposed, as primitive, structuring relations for creating non-predicative conceptual objects. The subject became that of what is thinkable, rather than of what was logically possible. Evidence for these epistemological primitives was taken from perception and ontology, with emphasis on what might be "natural" for a perceiving, plausibly reasoning system.

Since many of the network languages have been under constant development for years, they have generally admitted primitives at different times from each of several levels. The corresponding adjustments to interpretive mechanisms that must have followed cast some suspicion on the semantics of the languages as a whole, although in general, the mixing of levels has been found operationally useful (see [Hendrix, 1978], for example).

2.7 Pure epistemological languages

The move toward an epistemological basis for knowledge representation has begun only recently, and it is not yet clear where it will lead. It has not yet been determined whether a clear, formal semantics is possible. The research described in the second half of this paper is being conducted with the exploration of such a (non-logical) semantics in mind.*

Following the above discussion, then, it is to sets of **epistemological primitives** that we now turn. These are the primitives that constrain what an epistemological level language implicitly allows as the scope and structure of what can be thought. Such primitives are generally based around intensional descriptions of object types and relations between objects, as it

---

* See also [Levesque & Mylopoulos, 1978] and [Smith, 1978] for discussions of other pursuits of this goal.

seems that people are by their very nature locked into talk of objects (see [Quine, 1969], [Smith, 1978], and [Woods & Brachman, 1978]). The interesting thing here is that it is not necessarily the world that imposes objecthood, but instead it is a consequence of perception. Therefore, epistemological nets that support this ontology are not "semantic nets", since they say nothing about the world itself, but only about how it is possible to structure **perception** of the world (see [Fodor, 1978] for further discussion of the semantics side of this coin). That is, our enterprise, when building a net with object-type and relation primitives, is epistemology, and not semantics.

3. KLONE - An Epistemological Level Language

The quest for a set of epistemological primitives is based on two general requirements: 1) the need for a set of concept-structuring syntactic elements at a high enough level to make the factoring of general conceptual knowledge easy, perspicuous, and naturally reflect the way that we think about the world, and 2) the requirement that such a factoring still have a clear, precise, and formal semantics. Whereas logical languages like those described in [Schubert, 1976] and [Hendrix, 1975a,b] satisfy well the second requirement, there is some debate as to their amenability to "natural" conceptions of the world. For example, while predicate logic quantifier scoping is well-understood, and formally precise, it may be too "low-level" for expressing the kind of natural language quantification we might want to express (see [Van Lehn, 1978] for a discussion of how quantifier scope disambiguation may not be a natural process, but might instead be epiphenomenal). While the analysis of natural language quantifiers might be said to rest ultimately on a logical account, this might not be the best way to handle the representation of conceptualizations reflecting the meaning content of sentences using them. It is not the formal adequacy of logic that is at issue, but rather the level and appropriateness of its view of things. Further, the predilection of logical languages for predication may also be inappropriate: "We persist in breaking reality down somehow into a multiplicity of identifiable and discriminable objects..." [Quine, 1969, p. 1]. These objects have decomposable (conceptual) structure, as well as compositional integrity ("gestalt"). Quine believes that an ontology that allows attributes is potentially asking for trouble.

In contrast to logical level languages, those of the conceptual level seem to proffer structured conceptual objects - but at the expense of an unwritten pledge of allegiance to certain predetermined conceptual cases. Also, the semantics of such languages are suspect - or at least rarely discussed.

The intent here is not to argue that either the logical or conceptual approach is inadequate; rather, it is to hint that perhaps we should seriously consider the details of an intermediate approach - one that provides structured conceptual objects as its **raison d'etre**, but which does not presuppose any particular objects or case relations. Such is the substance of the "epistemological" level discussed in Section 2. The rest of this section is a look into the particular language that we are developing in an attempt to ferret out the structure and importance of this level of primitive. While research is continuing on this language - "KLONE" - it already contains a significant portion of the requisite representational apparatus. KLONE is currently implemented on a TOPS-20 system as a set of INTERLISP functions, which provide access to a KLONE data base (KLONE structures are implemented more or less as "abstract data types" - see Section 3.6).

3.1 Primary contributions of KLONE

KLONE is based on the idea of "Structured Inheritance Nets" [Brachman, 1978a,b], wherein descriptions of structured conceptual objects are organized in lattice-like networks, with inheritance between descriptions being carried by structured "cables".

There are two principal participants in concept structure: **Roles** and **Structural Descriptions.** A Role has internal structure which describes a conceptually identifiable subpart of a Concept; this structure includes a specification of the functional role that the subpart is to play in the conceptual complex, a description of the set of legal potential fillers for that role, and a connection back to the Concept itself. The Role is an attempt to capture the intensional description of a **role filler in context** - a place, for example, to hang information about the owner of a particular Volvo, even if the identity of that person is not yet known (or, if known, it is a place to separate statements about the person **as a car owner**, as distinct from statements about the person as a home owner, parent, etc.).

Structural Descriptions (SD's) take the set of conceptual subparts and express the structure of the interrelationships among them. These relationships give the Concept its "gestalt" - they, for example, differentiate between verbs whose case frames are identical, or express how a stack of bricks is different from a pile of the very same bricks.

This internal structure of Concepts gives the inheritance hierarchy its structured nature. Concepts, as structured wholes, can be related to other Concepts by **inheritance cables**, which pass the Roles and SD's from more general Concepts to others which subcategorize them. There are several ways in which a Role can be inherited, and since the number of Roles at a Concept is not constrained, there cannot be a single "ISA"-type link to carry the inheritance. Instead, the cable has (at least) one internal wire for each Role and each SD, specifying that Role or SD's particular relationship to its parent(s).

Since cables are self-contained, structured entities in and of themselves, KLONE can support a **multiple inheritance** mechanism. Each cable carries its own information about the superConcept and its Roles, and therefore multiple cables will not inadvertently interact. The explicit Role/SD inheritance avoids the problem of "slot" naming confusions common to most network formalisms. In addition, KLONE not only supports multiple independent superConcepts, but allows multiple Role parents and interrelationships between Roles inherited from different parents.

KLONE allows one to create "Individual" Concepts, that are descriptions of individual objects, and to relate such Concepts to **definitional**, "Generic" Concepts. Further, **assertional** connections are expressed by attachment of Individual Concepts to **Individual Constants**, which stand for previously identified (or potentially identifiable) extensional objects. This way the dichotomy between assertion and definition is kept clean, and the intensional world of the lattice is kept homogeneously descriptive.

The final critical feature of KLONE is its bipartite "attachment" mechanism. To any KLONE Concept, Role, SD, or cable, a **metadescription** can be attached. Metadescriptions treat entities **qua** entities, and express knowledge about knowledge in KLONE itself. In addition to the attachment of declarative representational structure, the attachment of procedures (interpretable by the KLONE interpreter) is a built-in feature. Such procedures are written in the language of the interpreter, and are invoked in particular situations, prespecified by the interpreter. Both kinds of attachment make KLONE networks well-suited for taxonomic (classificatory) operation, and the principal envisioned use of such nets is generalized "situation recognition" - see [Woods & Brachman, 1978].

I will now embark on a more detailed discussion of the particulars of KLONE epistemological representation.

3.2 Internal generic concept structure

Descriptions of structured conceptual objects appear in KLONE as **Generic Concepts.** Generic Concept has the appearance of a description of a single, PROTOTYPICAL member of a conceptual class, but in fact can be satisfied by an arbitrary number of differing, INDIVIDUAL descriptions of particular members. Generic Concepts act like intensional entities: they can be manipulated and reasoned about without regard to the extensional ("outside world") objects (if any) of the class described. Relations between Generic Concepts, and among their internal parts, are definitional - they make no claims about the "outside world", per se.

3.2.1 Roles

There are two varieties of epistemological subparts of KLONE Concepts: Roles and Structural Descriptions. These are discussed respectively in this subsection and the following one.

Roles represent the subpieces of the Concept that can be talked about in their own right - these express the "slottedness" of conceptual objects that is prevalent in "Frames", KRL units, MDS templates, case frames, etc. The Roles of a Concept express its perceived **generalized attributes**,* including its physical parts (if applicable), and other qualities important to the conceptual complex. Structurally, a Role interrelates three particular kinds of information:

1) the functional role that the conceptual subpiece will play vis-a-vis the Concept as a whole;**
2) a description of the particular kind of entity that can potentially fill that functional role;

---

* I have elsewhere referred to these as "**dattrs**" [Brachman, 1978b].

** The fact that a functional role is critical to this structure makes it a Role and not merely a "slot" (an implementation level view).

3) set-oriented information relevant when more than one entity can fill a given role.

A Role is unique to its associated Concept - descriptions of fillers and fillers themselves are independently identifiable Concepts, but Roles are not. This makes the Role an obvious place to tie information about a filler **in its context**, that is, information about some entity's participation in a structure, independent of its own internal structure - or even its existence. In Generic Concepts, Roles are intensional places to talk about agents of verbs, employees of companies, steering wheels of cars,* etc., **independent of any particular fillers.**

The current KLONE representation of Role structure is detailed in Fig. 1. A Role cannot exist without having some Concept as its context, and thus the epistemological connector called **RoleDef** must always be present in order to create a Generic Role.** The functional role information is captured by a set of pointers from the Concept's SD's (to be discussed in the next section), which express what a filler of this Role must do to be considered (in

---

* E.g., in "the alligator's tail fell off, but it grew back again", the "it" is referring to the **Role**, and not to the thing which filled it.

** In diagrams, I will use ovals for Concepts, squares for Roles, diamonds for SD's, arrows for epistemological connectors, and simple line segments for connections like RoleDef, one of whose participants cannot exist without being attached to the one on the other end.

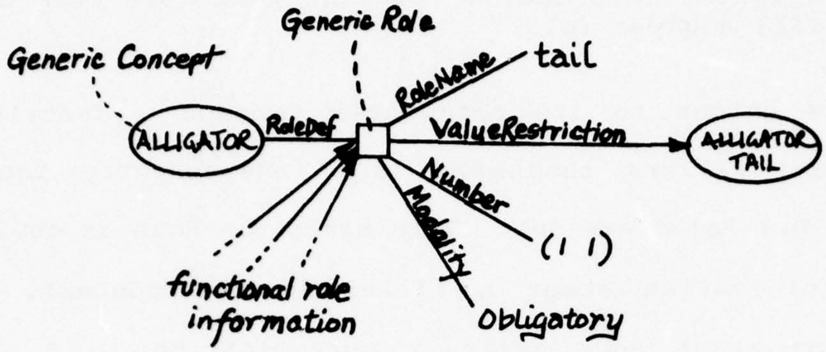

Fig. 1. KLONE Role structure.

this case) a **tail** of an ALLIGATOR. The **RoleName** connector allows an arbitrary, user-specifiable name to be associated with the complex of relationships associated with the Role by the set of pointers. RoleNames correspond to case labels in many notations, but here are not constrained by the language; in fact, RoleNames can be duplicated at the user's convenience, since they are only meaningful to him or her.

The **ValueRestriction** primitive ties a description of legal potential fillers to the Role. It points to another Generic Description, which must be satisfied by a Role filler in and of itself. Notice, then, that the ValueRestriction is a description that applies to a filler in its own right (i.e., describes its intrinsic characteristics), while the functional role information applies to any filler **by virtue of its involvement with other Concepts.**

Finally, the **Number** connected with a Role expresses the limitations on the cardinality of the **Set** of fillers of that Role. The current KLONE format allows only the specification of a range of values, with either the maximum or minimum (or both) possibly being "don't care". One could easily imagine a more general predicate as the number restriction (e.g., "even", "less than 12 or more than 50").

The epistemological Role structure provides a generalized "case" definition facility, when coupled with the Role inheritance to be discussed below. It offers an explicit handle on the intensional description of a filler of a role **for a particular Concept.** This is extremely valuable for Structured Inheritance (see Section 3.3).

3.2.2 Structural descriptions

While many of the conceptual subparts of an object can be reified, and talked about as "cases", "parts", or "attributes", there are relationships among these that are not referenced explicitly. The relationships are nevertheless critical to the Concept's description, as they provide the **meanings** for the cases, etc. Without such a structural "gestalt", for example, all verbs with identical case frames would be indistinguishable. The description of how the role players interact is as much a part of a Concept as are the descriptions of those players in and of themselves.

KLONE provides a set of epistemological structuring primitives for creating such relationship descriptions. Each Concept is allowed to have a set of Structural Descriptions, each of which comprises a complex of **Parameterized Individuals** (ParaIndividuals). A ParaIndividual is a copy of a Generic relational Concept, that is used in another to show how more than one role filler interrelates. The ParaIndividual (like the Role) is unique to the Concept (in fact, the SD) in which it appears.

Figure 2 illustrates detail of the concept structuring cluster. Within the single SD of the Generic Concept PEDESTAL is the ParaIndividual PI. PI is a version of SUPPORT that is unique to the Concept PEDESTAL - that is, any structural manipulations of the Generic Concept SUPPORT would affect all other Concepts that were in some way dependent on SUPPORT, but structural changes to the Parametric Individual PI would affect no Concepts dependent upon SUPPORT. In this figure, the relationship of the ParaIndividual (PI) to its defining Generic Concept (Support) is indicated by a **Structured Inheritance Cable** (see Section 3.2) labeled **"ParaIndividuates"**. This cable indicates that PI has the identical internal structure to SUPPORT, except that where SUPPORT

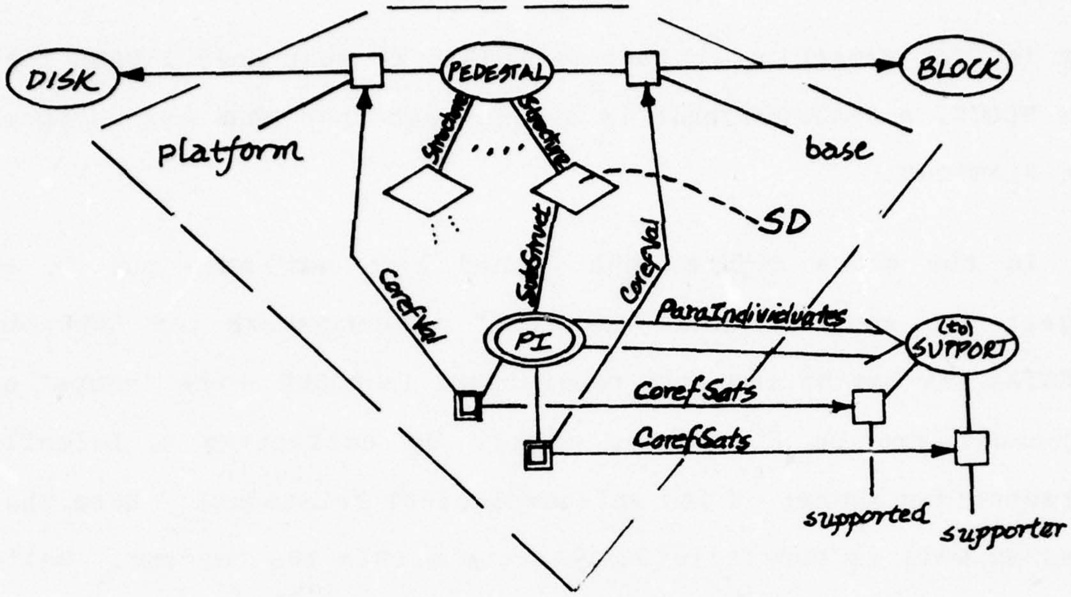

Fig. 2. A Simple SD.

has two Generic Roles, PI has instead two "Coref" Roles. Coref Roles stand in a similar relationship to Generic Roles as ParaIndividuals do to Generic Concepts. What the ParaIndividual and its Coref Roles mean is the following: in any particular individual which satisfies the description of the Generic Concept (in which the CorefRoles occur), there will be a particular instance of the Generic that the ParaIndividual comes from, relevant to that individual only. Each Coref Role will have its filler in that individual be coreferential with the filler of the particular Role indicated by the **CorefVal** link. For example, in any particular pedestal, P, there will be an individual SUPPORT relationship existing between P's own BASE, and P's own PLATFORM.

That is, for something to be a PEDESTAL, it must have a BASE that is a BLOCK, a PLATFORM that is a DISK, **and** that BASE must support that PLATFORM.

In the above figure, the dashed line has been put in to suggest the scope of the "internal" structure of the Concept, PEDESTAL (it has no explicit counterpart in KLONE - the "scope" of a Concept can be determined simply by collecting a formally characterized subset of its epistemological relations). Note that Roles as well as ParaIndividuals occur within the Concept. While Roles stand for single objects, and ParaIndividuals usually for relations, both are in a sense "parameterized" by the Generic Concept in which they occur. When an individual is described by that Concept, then its corresponding Role fillers and SD relations are firmly determined by the "template" structure in the Generic.

Roles and ParaIndividuals are similar in that each is a unique place to talk about an individual in a particular context. A unification of the two could be achieved by allowing a Concept merely to have an open-ended set of ParaIndividuals which can relate to each other. This would produce a structure akin to KRL's "perspectives", in which each use of a prototype (in our case, Generic) unit essentially results in a new version of the prototypes. Such a unification, however, yields a problematic (or at least uncomfortable) situation for slot (Role) inheritance (see below), and means that without proper care in structure - building, interrelationships between slots are either not accessible from all of the slots that they affect, or duplicated needlessly.* In addition, it does not seem evident that all aspects of a Concept's definition obtain equal ontological status, In particular, it is more useful to single out individual role players for naming than it is to reify and name every individual interaction among them.

3.3 Aspects of Structured Inheritance

With each KLONE Concept comprising a set of structured Roles and SD's, the notion of "inheritance" that is the mainstay of most network formalisms becomes a complex issue. To express the description of a Concept in terms of a more general one, we cannot merely connect the two with an "ISA" link** - we must account explicitly for how Roles and SD's of the more general Concept (the **superConcept**) are related to those of the less general (the **subConcept**). This is particularly critical when Concepts are allowed to have more than one superConcept: Roles "inherited" from one superConcept must not interact inadvertently with those from another.

---

* For more details on this problem, see [Brachman, 1978b], Section 8.2.1.

** See [Woods, 1975] and [Brachman, 1978b] for further discussion of ISA links.

The fact that Roles and SD's are explicitly addressable structures gives KLONE the leverage it needs to deal with inheritance in a clean, powerful way. The basic mechanism for inheritance is simple: a Concept inherits all Roles and SD's **intact** from each of its superConcepts through a **structured inheritance cable** (i.e., one cable per superConcept). With no altering structure at the subConcept, Roles and SD's of each superConcept are considered to exist as independent, identifiable entities, **even if RoleNames coincide.** The fact that Roles are not simply their names keeps any coincidental conflicts from arising, a problem that has plagued most network systems (see [Schneider, 1978], for example). KRL has eliminated this kind of "virtual copy" [Fahlman, 1978] inheritance by resorting to the unit/perspective difference, explicitly to avoid conflicts arising out of the fact that slots are only addressable by slot names.

Figure 3 illustrates the simplest cases of structured inheritance (superConcept links are represented by broad arrows). In Fig. 3(a), the cable C1 passes Roles R1 and R2 and SD's S1 and S2 from the Concept DOG to the Concept SCHNAUZER - to all interpreter routines accessing the Roles of SCHNAUZER, it appears as if that Concept is the direct owner of R1 and R2. That is, in this case of no modification of any of DOG's structures, the cable acts as a "short circuit" between the two Concepts, and SCHNAUZER

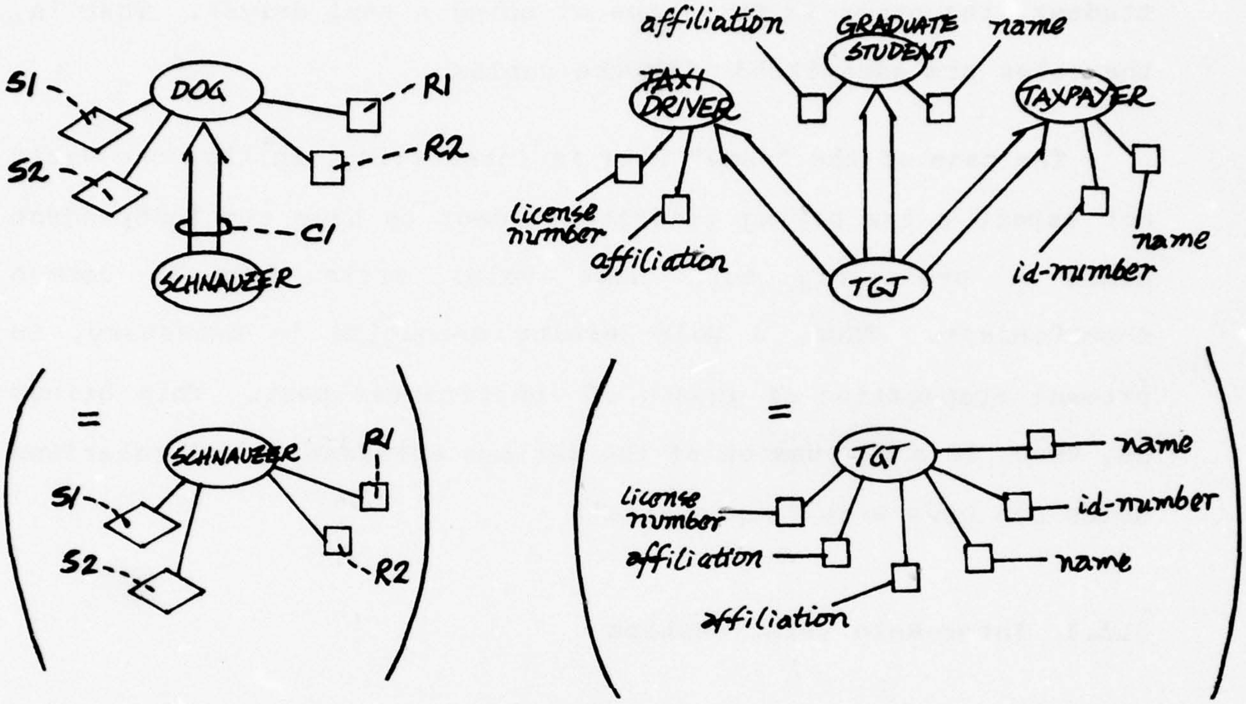

Fig. 3. Simple inheritance.

is virtually the same Concept as DOG (See [Fahlman, 1978]: also, note that this is, taxonomically speaking, a useless connection - there is no difference between the two Concepts). Figure 3(b) illustrates the same kind of inheritance for multiple superConcepts. To all Role-accessing processing routines, concept TGT looks as if it were the direct owner of six independent Roles, two of which happen to have the associated RoleName, "affiliation", and two of which happen to have "name" as their RoleNames. As far as KLONE is concerned, there is no connection between these. One affiliation of TGT is an affiliation by virtue of being a graduate

student, the other is by virtue of being a taxi driver. That is, the roles are associated with the *cables*.

The case of the "name" Role is interesting, in that one would not expect a tax paying graduate student to have two independent names - presumably both name Roles *arise from a common* superConcept. Thus, a Role-joining mechanism is necessary, to prevent propagation of ghosts of inheritances past. This brings us, then, to a discussion of the various epistemological relations Roles can have with "superRoles".

3.3.1 Inter-Role relationships

As suggested by the above examples, while Roles may be descended from those in super-Concepts, it is not always useful for a Role to be an exact copy of its source. In order to make subConcepts more specific descriptions than their superConcepts, we frequently need to modify the Role descriptions of the subConcept to make them more tightly constrained. In such a case, a Role is still "*inherited*", but some part or parts of it are to be altered in the process. Take, for example, the general Concept of a prepositional phrase, as illustrated in Fig. 4. Here, the Roles state that a PP is a tripartite structure, taking a PREPOSITION, a PREP-OBJECT, and a PP-OBJECT. Say, then, that we want to subcategorize PP into several more specific subtypes of PP - e.g., "TIME-PP" and" REGION-PP".

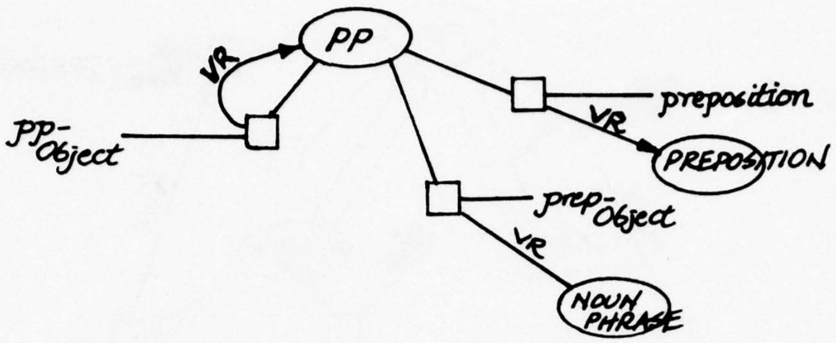

Fig. 4. Simple structure for a PP.

In each case, the restriction on the definition involves a tightening of the ValueRestriction to a more specific description than the general one at PP. Each subConcept of PP would, in KLONE, have an explicit Generic Role for each Role that was being modified, with an epistemological connector explicitly to the Role that it was changing. Thus, in Fig. 5, the component wires of the inheritance cables are indicated explicitly. These are labeled **Mods**, to indicate the type of change, in this case, an overriding of each piece of the Role that explicitly occurs in the subConcept. The meaning of Mods is that the indicated Role is inherited as a virtual copy, with each epistemological connector* (except RoleName) that appears explicitly "overwriting" the corresponding one inherited in the copy. Thus, the ValueRestriction for the REGION-PP's prep-Object Role is REGION-NP, while its Number is still (1 1), and its RoleName is unchanged.

---

* RoleNames are always inherited (although new ones can be added)

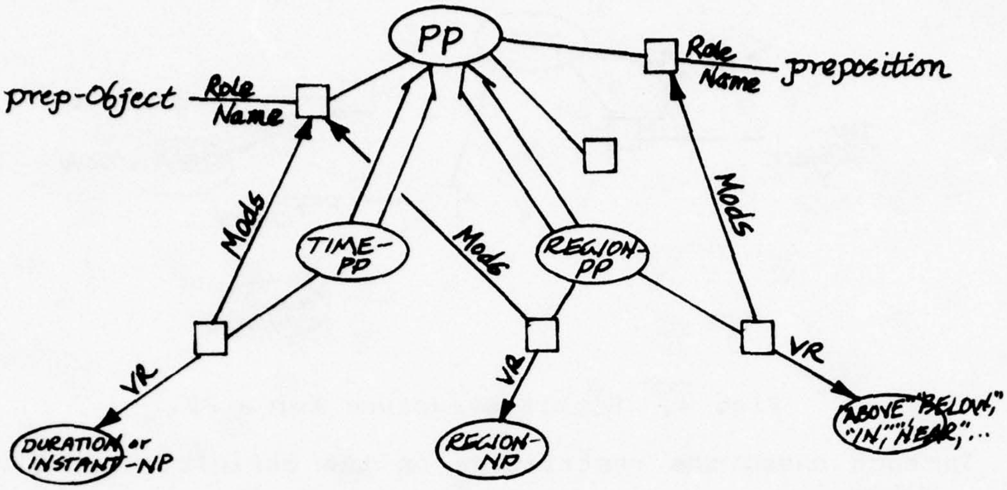

Fig. 5. Some Role modifications.

Another inter-Role relationship is that of ValueRestriction **satisfaction.** Satisfaction sets up a binding of a Role to a particular value; that is, it allows a Role to be filled. The Role-binding pair is captured in KLONE by a different kind of Role than the Generic, which intensionally represents a set of **potential** bindings. The Role is called an "Instance Role", and it simply represents the correspondence between the filler and its defining Generic Role. Figure 6 illustrates the use of Instance Roles to fix the values of certain parts of Generic descriptions. Once an Instance Role occurs in a chain of Concepts, it can no longer be modified (no epistemological connectors can attach to it from "below"). In Fig. 6, this means that any descendant of ON-PP **must**

---

since the fundamental identity of the role is not changing, even though its ValueRestriction might.

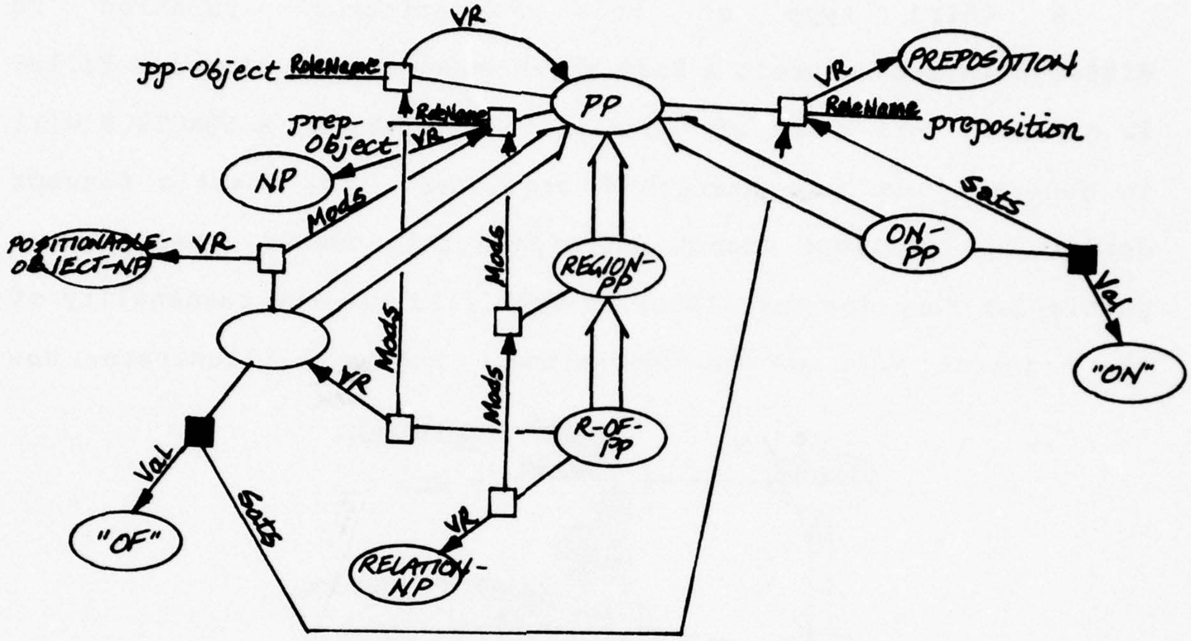

Fig. 6. Role satisfaction in Generics.

have the particular preposition, ON, as its preposition Role filler. In addition, an R-OF-PP can have as pp-Object only PP's whose prepositions are OF, and whose p-Objects are POSITIONABLE-OBJECT NP's (this covers cases like "to the north of England"). The two parts of Instance roles are indicated by **Sats** connectors to Generic Roles, and **Val** connectors to fillers.*

---

* Note that the Mods link between pp-Object Roles of R-OF-PP and PP skips the intermediate Concept, REGION-PP. By the normal mode of inheritance, the role from PP is for all intents and purposes a Role of REGION-PP as well, so R-OF-PP's Role can point directly to it. Thus, Cables work almost as if they were "conduits" for inheritance wires.

A third type of Role "inheritance" relation is **differentiation**, wherein a Role which expects more than one filler is expanded into a set of subRoles. For example, a FUNCTION will in general take some number of arguments. The Generic Concept describing functions cannot anticipate this number - only when a particular function (or class) is specified can the cardinality of the argument Role set be determined. Figure 7 illustrates how KLONE allows such a Role to be differentiated into three subRoles, each of which is an argument type. In the figure, the **Diffs** link stands for the epistemological differentiation primitive. The

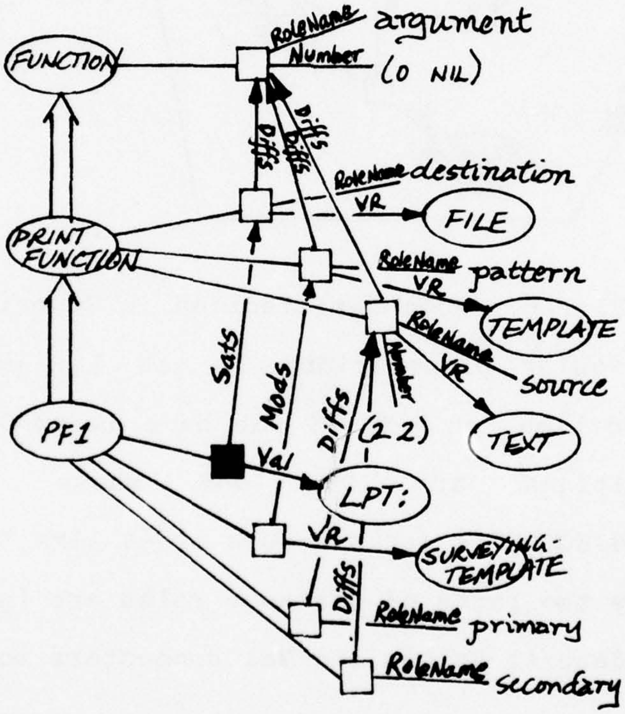

Fig. 7. Differentiating into subRoles.

Concept of a PRINTFUNCTION has three of its own Roles, each of which is expected to have one filler of the type specified as the ValueRestriction. Each of the source, pattern, and destination is still considered to be an argument, but now has a Role of its own.

Roles created by differentiating Generic Roles of a superConcept are normal Generic Roles, with their own Number and ValueRestrictions. They can be further modified, differentiated, or Satisfied, as Concept PF1 in Fig. 7 illustrates. Since the "source/argument" Role of PRINTFUNCTION indicates that it takes either one or two fillers, it can be further differentiated in "primary" and "secondary" sources (at this point, each's Number is forced to be (1 1)). Each of these has its own immediate RoleName, as well as "Source" and "argument", since a special kind of argument, for example, is nevertheless still an argument. The "pattern" Role of PRINTFUNCTION is modified in PF1 to have a more constrained ValueRestriction. Finally, an Instance Role that binds NIL to the destination/argument makes PF1 a type of PRINTFUNCTION whose destination is always the line printer.

3.3.2 Rules for single Role inheritance

The explicit connections between Roles and their subRoles are the "wires" running through the structured inheritance cables.

Each type of connector has a distinct meaning, and a formally specifiable set of inheritance rules. The following is a brief summary of the inheritance for Sats, Mods, and Diffs in the case of a Single Super Role:

Epistemological Primitive	SubRole Type	SuperRole Type	Number	ValueRestriction	RoleName
Sats	Instance	Generic	Always (1 1).	N/A	**All RoleNames of superRole.**
Mods	Generic	Generic	Take local if specified, else inherit superRole's Number.	**Take local if specified, else inherit superRole's** ValueRestriction.	**All RoleNames of superRole, plus** local.
Diffs	Generic	Generic	Take local if specified, else (1 1).	Take local if specified, else inherit superRole's ValueRestriction.	All RoleNames of superRole, plus local.

Sats expresses the filling of a Role, in which case, the Value Restriction of the SuperRole is expected to be satisfied by the filler, the Number of the SuperRole cannot be violated by its complete set of satisfiers, and the RoleNames applicable to the filler are taken from the defining SuperRole. Mods makes the SubRole a duplicate of the SuperRole - that is, it **is** the same Role - and therefore the Number and ValueRestriction from the parent are taken directly, unless overwritten with a local restriction. Diffs expands a general Role into more specific SubRoles; therefore, each SubRole is expected to play the part described in its Super Role, but does not supersede it (in the Mods case, it is as if the superRole has been absorbed completely by the subRole). The immediate effect of this is a default Number of (1 1). Notice that, in all cases, RoleNames apply universally down the inheritance chain - given the intent of Roles, one would not expect to try to change one Role into another in the middle of the lattice.*

3.3.3 Multiple Role inheritance

Recall our discussion of the case of the Graduate Student with two names (Fig. 3). The problem with inheriting Roles independently from several superConcepts is that several of them might be essentially the same Role. KLONE enforces a certain type of Role merging, and supports multiple superRoles in general.

The situation in which Roles are automatically merged is the following: whenever a new inheritance cable is added, if some subConcept would inherit more than one Role **descendant from a single superRole**, then form a new Role at that subConcept which jointly modifies each of those that would have been inherited

---

* One other way of introducing new Concepts in the lattice is by introducing completely new Roles (not descended from those of any superConcept) in the middle of an inheritance chain. This operation is useful, and is supported by KLONE, but its epistemological significance is hazy. We are currently investigating the utility and meaning of this type of modification.

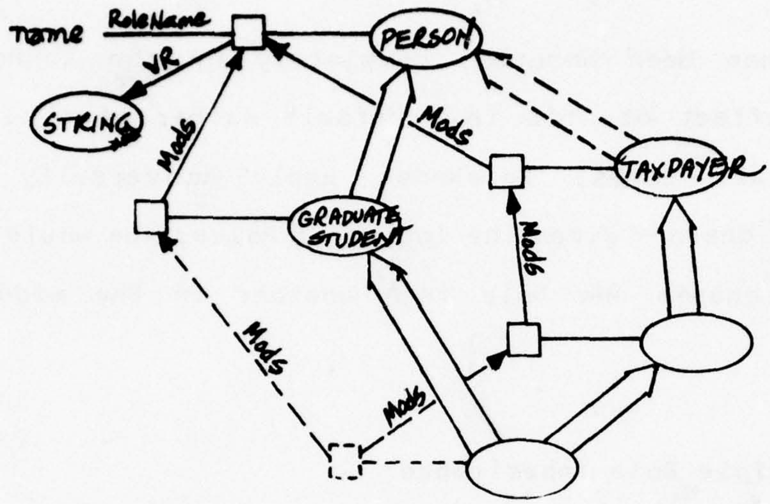

Fig. 8. Merging Roles.

independently. This merges two Roles, so that descendants of the subConcept will "see" only a single Role (see Fig. 8). The characteristics of that Role are the merged ones from its parents. These are calculated from the following simple rule: if a Role differentiates at least one other Role, then it acts as a subRole. Since its Number defaults to (1 1). Its ValueRestriction is a "conceptual intersection" of the Value Restrictions of all of its superRoles (the most general subConcept that covers all of those Concepts). In the other case, when all superRoles of a Role are modified, then the Role is simply a multiply overlaid virtual copy of all of the superRoles. In that case, all constraints are "intersected."

Note that having multiple superRoles is a strong merger - no longer are the individual ancestors of a Role available for manipulation in subConcepts. Other relations between Roles (including intensional equivalence) can be specified in SD's, so that roles that are inherited independently can begin to interrelate (the cables always serve to distinguish ancestry). This is a critically important combination of facilities, since multiple descriptions tend much more often to interact than to be independent (e.g. aspects of being a graduate student might motivate being a taxi driver, student id numbers are often the same as social security numbers, taxes must be paid on the wages earned as a taxi driver, etc.).

3.3.4 SD inheritance

KLONE currently supports only a simple form of SD inheritance. It is becoming clear that a mechanism comparable to that supported for Roles is needed, and this will be pursued in the coming year.

SD's are inherited intact through cables just as Roles are. There is a single modification primitive, through which an SD can preempt one that could have otherwise been inherited. When an SD modifies another in this way, the superSD is completely overridden - no ParaIndividuals from one are available to the other. This at least allows the modification of a Concept's inter-Role gestalt, but is obviously an oversimplified facility.

3.3.5 Meta-relations on structured inheritance cables

In a hierarchy of descriptions, there are relations between subConcepts of the same Concept that help support its use as a taxonomic classification. Certain descriptions can, by definition, exclude from their descendants the possibility of being multiply described by other subConcepts descended from the same parent. For example, the Concept PERSON may have two subConcepts, MALE and FEMALE. By definition, these may be expected to be mutually exclusive - no descendant of MALE can have a superConcept descended from FEMALE.

Mutual exclusion is not always the case, and in KLONE it must be marked explicitly where it applies. This is accomplished with a **mutual exclusion metadescription**, attached to a set of cables descending from a given Concept. Such a primitive prevents KLONE from ever building a joint subConcept descendant from any combination of those subConcepts whose cables are indicated. Figure 9 illustrates a split of the Concept PERSON along two dimensions (sex and age), and represents the fact that no MALE is a FEMALE, and no CHILD is a TEENAGER or an ADULT, etc. The absence of a statement connecting the sex subhierarchy and the age one means that multiple descriptions from both (in this case one from each) are allowed. This mechanism is similar to Hendrix's [1975a,b, 1978] distinction between "ds" and "s" links, but it

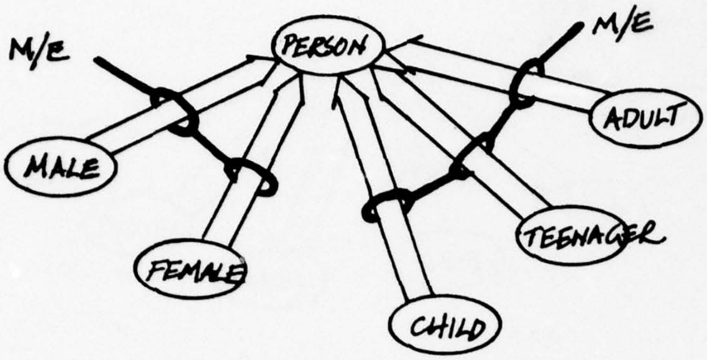

Fig. 9. Mutually exclusive descriptions.

allows more general n-ary relations without resorting to artificial intermediate subConcepts.

In a taxonomic structure, one might want to make essentially the opposite type of statement - that two types of description must apply to the same individual. KLONE supports this by allowing an **exhaustiveness metadescription** that makes the statement that certain subConcepts of a Concept exhaust the space described by that Concept. For example, Fig. 10 expresses the fact that rear-wheel drive (RWD) cars, front-wheel drive (FWD) cars, and four-wheel drive (4WD) cars exhaust the space of types of cars. This allows KLONE to determine that any FORD, VOLVO, PLYMOUTH, etc. **must** be one of these three types of cars.

Augmenting the structure of Fig. 9 to include exhaustiveness constraints in the same places as the mutual exclusion ones yields a structure that expresses two **partitions** of the space of PERSONS:

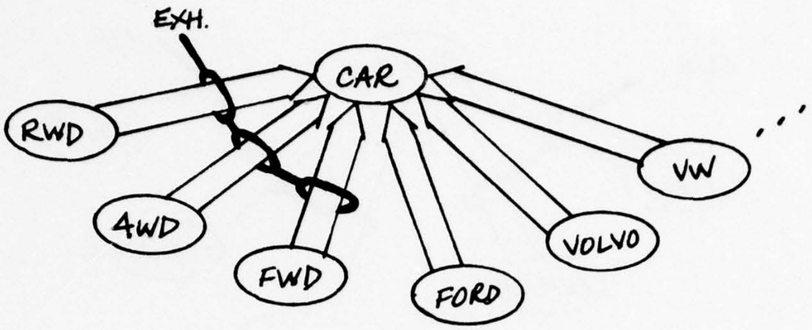

Fig. 10. Subdescription exhaustiveness.

MALE/FEMALE, and CHILD/TEENAGER/ADULT. Any CHILD must be either a MALE or a FEMALE, and cannot be either a TEENAGER or an ADULT, etc.

3.4 Individuals and individuation

Generic Concepts in KLONE, as stated earlier, are descriptions that are to be applied to individuals. One can join many superConcepts to make up a complex description - one that may, in fact, be so complex that it could be considered to apply to only one individual in the universe. This, in some sense, allows us to "multiply describe" an individual in the world outside of the machine, and appears to allow us to focus on a single individual. But neither is quite the case.

Individuals, and descriptions of them, are generally problematic in knowledge represention languages. In KLONE we have chosen to distinguish carefully between the internal conception of an individual object (outside of the machine) and descriptions of individuals. A KLONE entity called an **Individual Constant** is created to stand directly for some object (concrete or abstract) when it has been concluded that one exists. This type of representational entity has no internal structure as far as KLONE is concerned. An Individual Constant is merely a place to attach descriptions that are meant to refer to the same real-world object. That is, the Individual Constant, while abstractly "standing for" some real entity,* is concretely for the machine a **locus of description coreference.**

The potential causes of the creation of an Individual Constant are not really the concern of the representation language, but one might immediately envision two: first, the reasoning of the machine after some natural language input might cause it to come to believe that an object that is being referred to exists (see [Webber, 1978]) - that is, it might be **told** that there is someone out there. This could occur in discussions of hypotheticals, and there is probably a relevant context mechanism that KLONE does not yet account for. Second, the machine's perceptual processes (especially visual and touch information) might cause it to decide

---

* The realness of one of these entities is strictly in the eye of the beholder. "Real-world objects" are what the machine believes to be outside of itself, and there is no contention here that anything in fact does exist there.

that it is in contact with some object. Thus an Individual Constant could be constructed on non-introspectable grounds (i.e., not by virtue of a description). This has potential for being the *interface between the machine's intellectual and sensorimotor* components (see [Becker, 1972]).

One could imagine creating an Individual constant to reflect a belief in an object, and then attaching descriptions of various sorts to it. Fig. 11 illustrates how one might attach to the Individual Constant "DAH" (drawn as a shaded hexagon) the two descriptions, "a person with red hair", and "an employee of BBN". The wavy lines indicate the epistemological primitive of description-assertion. Note that these connectors are strictly assertional, and may be subject to negation when new data arrives.

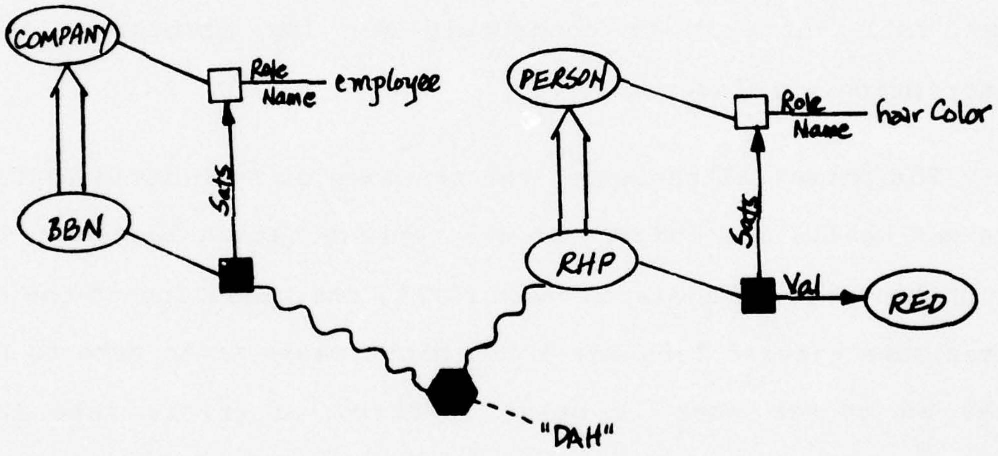

Fig. 11. Asserting descriptions.

3.4.1 Individual Concepts

An Individual Concept looks and behaves much like a Generic Concept. It inherits Roles in the same manner, through an **Individuation** cable, which is similar to the normal superConcept cable. However, an Individual Concept is required to be uniquely determined by its superConcept plus the specified values of its roles. Another restriction on Individual Concepts is that they cannot be further subcategorized. They are "leaves" hanging off the bottom of the lattice. Finally, the transition from Parametric Individuals to real, individual relationships is made through the individuation cable. An Individual Concept inherits not just the SD's as specified in its parent Generic, but fully individuated versions of them with Instance Roles taking the place of Coref Roles, and the corresponding values for the individual filling those Instance Roles.

Consider Fig. 12 as a simple example (where individuation is indicated by the broad solid arrow). This network fragment expresses the belief that there is one individual in the world (labeled "ARC-DE-TRIOMPHE"), and that individual is multiply described as "a MONUMENT built by NAPOLEON" and "the 75' ARCH in PLACE-DE-LA-CONCORDE." The Individual Concept for the latter (A1) implies that if any other ARCH is found with the same values for its Roles, then that ARCH must be the ARC-DE-TRIOMPHE. If the

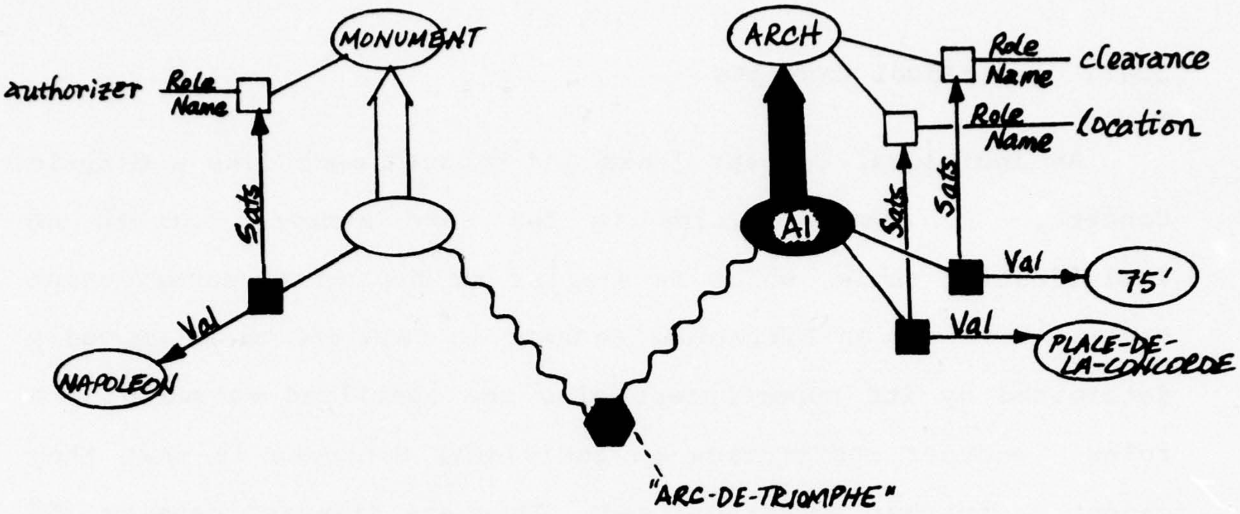

Fig. 12. An Individual Concept.

machine were to encounter a monument built by Napoleon, it could not conclude from the preceding description that it was the same arch. Thus, Individual Concepts become very important for reasoning within the ontology of objects. If one were to find that the same Individual Concept described two different Individual Constants, he would be forced to conclude either that what he thought were two individuals was in fact one, or that one of the descriptions is wrong.

3.5 Metadescriptions and ihooks

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 Brian Smith [1978], the epistemological import of "procedural attachment" is now much clearer. There are, according to Smith, **two** different types of attachment that are most often confused under the guise of "procedural attachment": 1) "meta-description", wherein knowledge about knowledge is expressed in the same language as the primary knowledge (e.g., KLONE itself); 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 meta-description, the interpreter is being asked to make a type or level jump when processing a Concept* Meta-information 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 in general, KLONE provides an epistemological primitive that ties a metalevel Individual Constant to a formal entity in the network. It allows us to talk about that entity as a "real" individual. This link is

---

* This is the case in the Levesque and Mylopoulos [1978] "instance hierarchy".

called a "metahook", and it can attach to a Concept, a Role, or an SD. Descriptions of various sorts can then be attached to the Constants, thereby describing the individual formal entities.

Metadescriptions can be used for various purposes. One use of them in the current KLONE system is to express "default" fillers for Roles. Smith [1978] points out that default information is basically advice to the reasoning mechanism, rather than basic epistemological structure. This is reflected in KLONE by the kind of structure illustrated in Fig. 13. D is an assertion holding between R and C, expressing that C is the value to be used to fill R when no other can be found (at the moment, the interpreter knows the Concept DEFAULT, but ultimately it will be expressed in terms

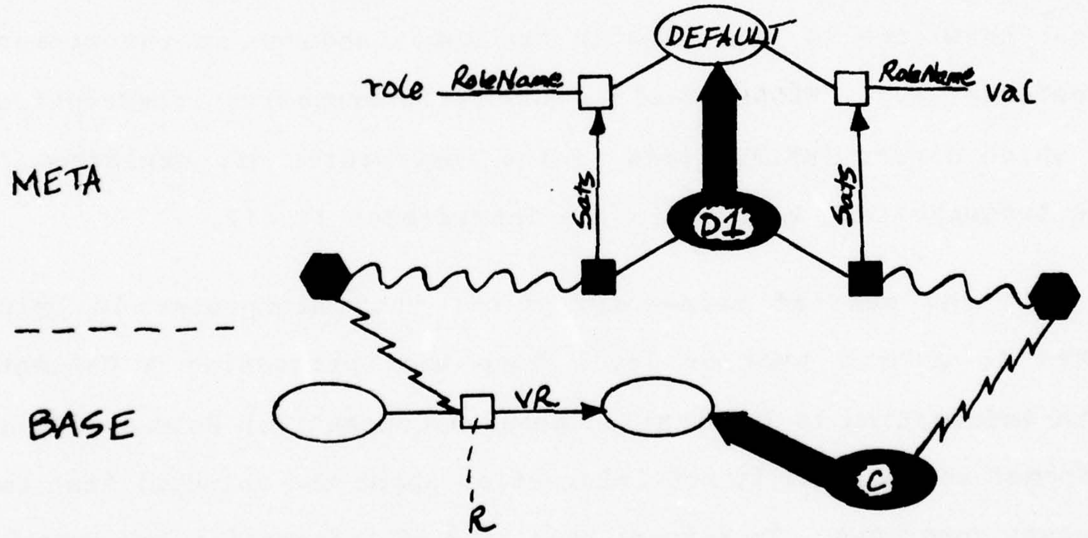

Fig. 13. A metadescription for a DEFAULT.

of other Concepts). Notice that with defaulting being metadescribed, it is possible to specify a functionally determinable value instead of a single fixed one, as a default. A description of where to find that value would be expressed in KLONE itself - the "val" Role of the DEFAULT Concept (D) would be filled by a structured object - a Concept - rather than tied by an assertion connector to an Individual Constant.

KLONE provides another kind of hook, the "interpretive hook" (ihook), for attaching interpreter code directly to a Concept, Role, or SD. 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 rather as a means of direct advice to the interpreter with clear import.* The code pointed to by an ihook must be constructed from interpreter primitives (e.g., functions like "CreateConcept", "SatisfyRole", etc.), and the ihook must specify the place in the interpreter from which the code is to be invoked.

Ihooks have the following annotation structure:

<Basic invocation type Situation-description Keyword>.

---

* See [Smith, 1978] for details on the interaction of interpretive "planes" and meta-descriptive "layers".

The basic invocation type is either "PRE-", "POST-", or "WHEN-". Each interpreter function, before it executes its main body of code, evaluates as preconditions any applicable procedures whose ihooks start with "PRE-". If any procedure fails, the function returns immediately, and never alters any structure. Upon success of the pre-conditions, the function does what it is supposed to, and then evaluates the "POST-" conditions. Again, if any condition fails, the entire operation is undone, and the function aborts. Finally, if the post-conditions are met, "WHEN-" procedures are run to perform side-effects.

Applicability of attached procedures is determined by the current situation and keyword. Situation descriptions are simply "individuating", "modifying", "changing", etc. - one corresponding to each operation that can be done on an epistemological primitive. These are organized into a simple hierarchy, so that certain sets of functions can trigger the same procedure. Keywords are user-specifiable and arbitrary. They allow a crude form of inheritance overriding for ihooks: a Concept, Role, or SD inherits the set of procedures from all of its parents. However, only one procedure with a given keyword - the most specialized one - is evaluated.

3.6 Interpreting KLONE structures

The impression I have given that there is a single KLONE interpreter that deals with the epistemological primitives described here is a bit misleading. KLONE is implemented (in INTERLISP) as a set of interpreter primitives. Together these 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 necessary access to the KLONE structures. In this way, Concepts, Roles, and SD's are abstract data types as 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 we do not yet 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. 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 SD's is fixed in advance. In order to implement, say, a function that finds a (possibly inherited) restriction for 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 Role inheritance functions, including the provision for multiple superConcepts and multiple superRoles, has been implemented based on the small set of possible inter-Role relationships.

Since the users of KLONE "see" only abstract structures for Concepts, etc., it is not necessary for them to think of the network as a set of nodes interconnected by links. Instead they can view Concepts as sets of Roles and SD's. The functions deal only with those entities (and their "epistemological" relations), and never attempt to make or break simple local link-like connections. This is important considering that **structured** inheritance, by means of a cable that contains many connections that are not independent, is a centered feature of KLONE. Thus, cables are KLONE's solution to one problem with the traditional semantic network metaphor in general: the apparent independence of each link from all other links.

4. Continuing Research on KLONE

In the second year of this project, we intend to continue research into representation issues raised by the KLONE epistemological paradigm. In addition to pursuing the particular issues raised below, we plan to fully investigate the system with a parser, develop general inference algorithms (including "situation recognition"), include a discourse structure for anaphora resolution and speech act identification, and catalog the "common sense" Concepts of the research domain.

4.1 Issues in Roles

Role structure, as it currently stands, does not account for the complete set of desirable behaviors for Roles. While it seems to make sense that a Role is never shared between Concepts, there are cases that indicate needs in that direction. For example, some Roles should probably allow specification of at least RoleName and Number in the reverse direction. While any Concept has access to all Roles in which it participates, (i.e., the epistemological connectors can be traversed in either direction), it cannot seek out **named** relations without itself having explicit Roles for them. For example, if I fill the **employee** Role of a COMPANY Concept with a description of some person, I am forced to give PERSON a Role called **employer** in order to access that inverse functional role by name. It would be nice if **employer** were defined merely to be the inverse of **employee**. KLONE should probably provide an inverse role capability, so that such a Role can be accessed naturally.

Further, we might want to limit a Concept's participation in a given Role (or set of Roles), by specifying an "inverse Number" restriction.* At the moment, KLONE provides no way to specify that a given filler is limited in its participation in other Roles of a Concept.

---

* This idea derives from the work of Levesque and Mylopoulos [1978], wherein a "structural attribute" (Role) is an instance of

Going one step further, one might even imagine a kind of Role that is very close to being shared between two Concepts. Such would be the case when two Concepts had "inverse" Roles - when one was filled, the other would be filled (with the opposing Concept). KLONE has no mechanism for enforcing automatic filling of such opposite Roles. The underlying epistemological structure in this case might just be something close to two Concepts sharing a Role, each looking on it from the opposite point of view.

Finally, given that KLONE allows multiple fillers of a given Role, there probably should be access to the **set** of fillers as a separable entity from the fillers themselves. As KLONE now stands, there is no way to state that a Role with a Number range has been filled as many times as it ever will be; that is, one cannot yet say that the set of fillers is complete. That is a statement about the cardinality of the Role filler set, as a set. We can also imagine a case wherein we might like to express that the set of fillers is the same as some other set, although the latter's members aren't currently known. Again, this is evidence for the need to address the role fillers as a set,* and we are currently

---

a **relation,** with domain and range each having a set of restrictions.

* The way that this problem has been avoided elsewhere [Bobrow & Winograd, 1977] is by assuming that Roles always have one filler and that the filler can be a set. We feel that this is not a tenable position, since it forces one to create an artificial intermediate structure, and to place an artificial burden on slots.

investigating a change to the representation of Roles that would facilitate talking about Role sets. Fig. 14 illustrates our thought on the subject: Roles (including Instance Roles) would still be direct subparts of Concepts, and an extra Conceptual piece - the role set - would be available for various kinds of manipulations.

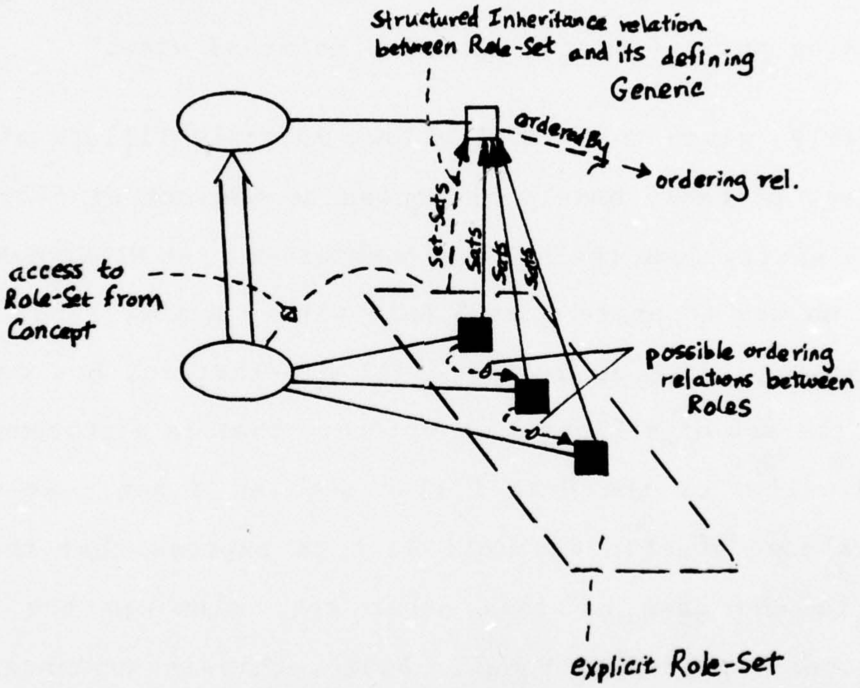

Fig. 14. A possible representation for Role-Sets.

4.2 Set descriptions

Sets are becoming increasingly important in KLONE. As mentioned above, sets of Roles will probably soon have explicit places in the epistemology. In addition, with the treatment of natural English discourse on the horizon, we must be prepared to deal with sets as they are used in anaphora and natural language quantification.* Webber's [1978] work has established some important requirements for handling anaphora candidate selection, and sets play a critical role in her analysis. For example, in "I *gave each girl a green tie-dyed* T-shirt. I bought **them** at Filene's," the "them" refers to an implicit set of T-shirts, never mentioned in the first sentence.

Our plan is to provide KLONE with a powerful set description facility to accommodate as many cases of intensional sets as possible. This facility would be augmented by an explicit set mechanism, to allow complex set descriptions to be made (e.g., "Ron, Brian, Marilyn, and three people on the canoe trip who knew at least two other people"). Sets will most likely occur in KLONE as different types of objects from those described here in this report (i.e., we do not expect to use, except at the metalevel, the Concept, SET), and will have different sorts of epistemological

---

* See Van Lehn [1978], for a treatment of natural quantifier scope determination.

subparts than Concepts. We also expect a close tie between set descriptions and Concepts - any Concept has implicitly associated with it the set of all things that satisfy its description. Originally, we had thought of leaving this as the only type of set, but it has recently become clear that set description by the characteristic function of a Generic Concept is too limited to handle natural language information. We are currently considering explicit sets fully specified ({a,b,c}), explicit sets incompletely specified ({a,b,c,...}), sets described by their own intrinsic features (e.g., cardinality), sets formed by **describing** their members, "maxsets" (the set of all x's), and sets derived by mappings from other sets. KLONE's structural power makes the last of these an extremely promising area for research.

4.3 Modality

In the past, we have associated with Roles an epistemological connector designating the "Modality" of the Role (see Fig. 1). Modalities included **Optional**, **Obligatory**, and **Inherent**, and were an attempt to designate which Roles had to be filled in order to consider a description as satisfied. As it turns out, this use of the Role structure to some extent confused definitional information with recognition-oriented information. These, it appears, are best left distinct, since one has to do with the basic representational quality of the formalism, and the other with its use. In our quest for cleanliness and clear epistemological grounds for language constructs, we are rethinking this facet of Role structure. It interacts strongly with the Number constraints (a minimum of zero fillers is as good as saying the filler is optional) and the derivability of fillers from SD specifications. We hope to have this understood within the coming year.

5. References

Becker, Joseph. 1973. "Robot" Computer Problem Solving System. Final Progress Report. BBN Report No. 2646. Cambridge, MA: Bolt Beranek and Newman Inc., September 1973.

Bobrow, Daniel G. 1977. "KRL, A Knowledge Representation Language." In **Proceedings of the Fifth International Joint Conference on Artificial Intelligence**, Cambridge, MA, August 1977, pp. 983-984.

Bobrow, Daniel G., and Winograd, Terry. 1977. "An overview of KRL, a knowledge representation language." **Cognitive Science**, Vol. 1, No. 1, January, 1977, pp. 3-46.

Brachman, Ronald J. 1977. "What's in a concept: Structural foundations for semantic networks." **International Journal of Man-Machine Studies 9**. March, 1977, pp. 127-152. Also BBN Report No. 3433. Cambridge, MA: Bolt Beranek and Newman Inc., October 1976.

Brachman, Ronald J. 1978a. "On the epistemological status of semantic networks." To appear in **Associative Networks -- The Representation and Use of Knowledge in Computers.** Nicholas V. Findler, **ed.** New York: Academic Press. Also BBN Report No. 3807. Cambridge, MA: Bolt Beranek and Newman Inc., April 1978.

Brachman, Ronald J. 1978b. "A Structural Paradigm for Representing Knowledge," Ph.D. dissertation, Division of Engineering and Applied Physics, Harvard University. Also, BBN Report No. 3605. Cambridge, MA: Bolt Beranek and Newman Inc., May 1978.

Carbonell, Jaime R. 1970a. "Mixed-initiative man-computer instructional dialogues." BBN Report No. 1971. Cambridge, MA: Bolt Beranek and Newman Inc., May 31, 1970.

Carbonell, Jaime R. 1970b. "AI in CAI: An artificial intelligence approach to computer-aided instruction." **IEEE Transactions on Man-Machine Systems**, Vol. MMS-11, No. 4, December 1970.

Cercone, Nick. 1975. "Representing natural language in extended semantic networks." Technical Report TR75-11. Edmonton, Alberta: Department of Computing Science, The University of Alberta, July, 1975.

Cercone, Nick, and Schubert, Len. 1975. "Toward a state based conceptual representation." In **Proceedings of the Fourth International Joint Conference on Artificial Intelligence,** Tbilisi, Georgia, USSR , September, 1975, pp. 83-90.

Fahlman, Scott E. 1977. "A system for representing and using real-world knowledge." Ph.D. Thesis Draft. Cambridge, MA: Artificial Intelligence Laboratory, MIT, June 15, 1977.

Fikes, Richard E. and Hendrix, Gary G. 1977. "A Network-based Knowledge Representation and Its Natural Deduction System." In Mechanical Intelligence: Research and Applications, S.R.I. Report No. 4763.

Fodor, J.A. 1978. "Tom Swift and his Procedural Grandmother", unpublished MIT mimeo.

Goebel, Randy and Cercone, Nick. 1978. "Representing and Organising Factual Knowledge in Proposition Networks." In **Proceedings of the Second National Conference Canadian Society for Computational Studies of Intelligence,** pp. 55-61.

Grosz, Barbara J. 1977. "The representation and use of focus in a system for understanding dialogue." In **Proceedings of the Fifth International Joint Conference on Artificial Intelligence,** Cambridge, MA, August, 1977, pp. 67-76.

Hayes, Philip J. 1977a. "Some association-based techniques for lexical disambiguation by machine." TR25. Rochester, NY: Computer Science Dept., The University of Rochester, June 1977.

Hayes, Philip J. 1977b. "On semantic nets, frames and associations." In **Proceedings of the Fifth International Joint Conference on Artificial Intelligence,** Cambridge, MA, August 1977, pp. 99-107.

Hays, David G. 1973. "Types of processes on cognitive networks." Paper presented at the 1973 **International Conference on Computational Linguistics,** Pisa, Italy, August 27-September 1, 1973.

Hendrix, Gary G. 1975a. "Partitioned networks for the mathematical modeling of natural language semantics." Technical Report NL-28. Austin, TX: Dept. of Computer Science, The University of Texas at Austin, December 1975.

Hendrix, Gary G. 1975b. "Expanding the utility of semantic networks through partitioning." In **Proceedings of the Fourth International Conference on Artificial Intelligence**, Tbilisi, Georgia, USSR, pp. 115-121.

Hendrix, Gary G. 1976. "The representation of semantic knowledge." In **Speech Understanding Research: Final Technical Report.** Donald E. Walker, **ed.** Menlo Park, CA: Stanford Research Institute, October 1976.

Hendrix, Gary G. 1978. "Encoding knowledge in partitioned networks." To appear in **Associative Networks -- The Representation and Use of Knowledge in Computers.** Nicholas V. Findler, **ed.** New York: Academic Press.

Levesque, Hector J. 1977. "A procedural approach to semantic networks." Technical Report No. 105. Toronto: Dept. of Computer Science, University of Toronto, April, 1977.

Levesque, Hector, J., and Mylopoulos, John. 1978. "A procedural semantics for semantic networks." To appear in **Associative Networks -- The Representation and Use of Knowledge in Computers.** Nicholas V. Findler, ed. New York: Academic Press.

Liskov, Barbara and Zilles, Stephen. 1975. "Specification Techniques for Data Abstractions." In **IEEE Transactions on Software Engineering**, Vol. SE-1, No. 1, March 1975.

McDermott, Drew. 1978. "Last Survey of Representation of Knowledge." In **Proceedings AISB Conference**, Hamburg, Germany, 1978.

Martin, William A. 1977. "OWL." In **Proceedings of the Fifth International Joint Conference on Artificial Intelligence**, Cambridge e, MA, August, 1977, pp. 985-987.

Nash-Webber, Bonnie L. and Reiter, Raymond. 1977. "Anaphora and Logical Form: On Formal Meaning Representations for Natural Language." In **Proceedings of the Fifth International Joint Conference on Artificial Intelligence**, Cambridge, MA. August, 1977, pp. 121-131.

Norman, Donald A. 1972. "Memory, knowledge, and the answering of questions." CHIP Technical Report 25. La Jolla, CA: Center for Human Information Processing, University of California, San Diego, May 1972.

Quillian, M. Ross. 1966. "Semantic memory." Report AFCRL-66-189. Cambridge, MA: Bolt Beranek and Newman Inc., October, 1966.

Quillian, M. Ross. 1967. "Word concepts: A theory and simulation of some basic semantic capabilities." **Behavioral Science**, Vol. 12, No. 5, September, 1967, pp. 410-430

Quillian, M. Ross. 1968. "Semantic memory." In **Semantic Information Processing**. Marvin Minsky, ed. Cambridge, MA: the MIT Press, pp. 227-270.

Quillian, M. Ross. 1969. "The Teachable Language Comprehender: A simulation program and theory of language." **Communications of the Association for Computing Machinery**, Vol. 12, No. 8, August, 1969, pp. 459-476.

Quine, Willard Van Orman. 1969. **Ontological Relativity and Other Essays**. New York: Columbia University Press.

Reiter, Raymond. 1978. "On Reasoning by Default." In **Theoretical Issues in Natural Language Processing-2**, pp. 210-218.

Roberts, R. Bruce, and Goldstein, Ira P. 1977. **FRL Users' Manual**. A.I. Memo No. 408. Cambridge, MA : Artificial Intelligence Laboratory, MIT, April, 1977.

Schank, Roger C. 1972. "Conceptual dependency: A theory of natural language understanding." **Cognitive Psychology, 3**, pp. 552-631.

Schank, Roger C. 1973a. "The conceptual analysis of natural language." In **Natural Language Processing**. Randall Rustin, ed, New York: Algorithmics Press, pp. 291-309.

Schank, Roger C. 1973b. "Identification of conceptualizations underlying natural language." In **Computer Models of Thought and Language**. Roger C. Schank and Kenneth Mark Colby, eds. San Francisco: W.H. Freeman and Co., pp. 187-247.

Schank, Roger C. 1977. "Conceptual Dependency, and Knowledge Structures." In **Proceedings of the Fifth International Joint Conference on Artificial Intelligence**, Cambridge, MA, August 1977, pp. 988-989.

Schank, Roger C., and Rieger, Charles, J. III. 1974. "Inference and the computer understanding of natural language." **Artificial Intelligence**, Vol. 5, No. 4, Winter 1974, pp. 373-412.

Schneider, P.F. 1978. "Organization of Knowledge for a Procedural Semantic Network Formalism." In **Proceedings of the Second National Conference, Canadian Society for Computational Studies of Intelligence**, pp. 81-89.

Schubert, L.K. 1976. "Extending the expressive power of semantic networks." **Artificial Intelligence**, Vol. 7, No. 2, Summer, 1976, pp. 163-198.

Shapiro, Stuart C. 1971a. "The MIND system: a data structure for semantic information processing." Technical Report R-837-PR. The Rand Corporation, August 1971.

Shapiro, Stuart C. 1971b. "A net structure for semantic information storage, deduction, and retrieval." in **Proceedings of the Second International Joint Conference on Artificial Intelligence**, pp. 512-523.

Shapiro, Stuart C. 1975. "An introduction to SNePS." Bloomington: Computer Science Department, Indiana University, March, 1975.

Shapiro, Stuart C. 1977. "Representing and locating deduction rules in a semantic network." SIGART Newsletter, No. 63, June, 1977, pp. 14-18.

Shapiro, Stuart C. 1978. "The SNePS semantic network processing system." To appear in **Associative Networks - The Representation and Use of Knowledge in Computers.** Nicholas V. Findler, **ed.** New York: Academic Press.

Simmons, Robert F. 1973. "Semantic networks: Their computation and use for understanding English sentences." In **Computer Models of Thought and Language.** Roger C. Schank and Kenneth Mark Colby, **eds.** San Francisco: W.H. Freeman and Co., pp. 63-113.

Simmons, Robert F., Burger, John F., and Schwarcz, Robert M. 1968. "A computational model of verbal understanding." **AFIPS Conference Proceedings,** Vol. 33 (1968 Fall Joint Computer Conference), pp. 441-456.

Simmons, Robert F., and Bruce, Bertram C. 1971. "Some relations between predicate calculus and semantic net representations of discourse." In **Proceedings of the Second International Joint Conference on Artificial Intelligence,** pp. 524-529.

Smith, Brian C. 1978. "Levels, layers, and planes: The framework of a system of knowledge representation semantics." Master's thesis. Cambridge, MA: Artificial Intelligence Laboratory, MIT, January 1978.

Sridrahan, N.S. 1978. **AIMDS User Manual - Version 2.** Department of Computer Science, Rutgers University, CBM-TR-89.

Srinivasan, Chitoor V. 1976. "The Architecture of Coherent Information System: A General Problem Solving System". In **IEEE Transactions on Computers**, Vol. C-25, No. 4, April 1976, pp 390-402.

Szolovits, Peter, Hawkinson, Lowell B., and Martin, William A. 1977. "An overview of OWL, a language for knowledge representation." MIT/LCS/TM-86. Cambridge, MA: Laboratory for Computer Science, MIT, June, 1977.

Van Lehn, Kurt. 1978. **Determining the Scope of English Quantifiers.** MIT Artificial Intelligence Laboratory, Technical Report No. 483.

Webber, Bonnie L. 1978. **A Formal Approach to Discourse Anaphora,** Ph.D. dissertation, Harvard University, 1978. Also BBN Report No. 3761, Cambridge, MA: Bolt Beranek and Newman Inc., 1978.

Winograd, Terry. 1978. "On primitives, prototypes, and other semantic anomalies." In **Theoretical Issues in Natural Language Processing-2,** pp. 25-32.

Winston, Patrick H. 1970. "Learning structural descriptions from examples." Project MAC TR-76. Cambridge, MA: MIT.

Woods, William A. 1975. "What's in a link: Foundations for semantic networks." In **Representation and Understanding: Studies in Cognitive Science,** Daniel G. Bobrow and Alan M. Collins, **eds.** New York: Academic Press, pp. 35-82.

Woods, William A. and Brachman, Ronald J. 1978. Research in Natural Language Understanding, Quarterly Technical Report No. 1, 1 September 1977 to 30 November 1977, BBN Report No. 3742. Cambridge, MA: Bolt Beranek and Newman Inc., January 1978.

Official Distribution List
Contract N00014-77-C-0371

	Copies
Defense Documentation Center Cameron Station Alexandria, VA 22314	12
Office of Naval Research Information Systems Program Code 437 Arlington, VA 22217	2
Office of Naval Research Code 200 Arlington, VA 22217	1
Office of Naval Research Code 455 Arlington, VA 22217	1
Office of Naval Research Code 458 Arlington, VA 22217	1
Office of Naval Research Branch Office, Boston 495 Summer Street Boston, MA 02210	1
Office of Naval Research Branch Office, Chicago 536 South Clark Street Chicago, IL 60605	1
Office of Naval Research Branch Office, Pasadena 1030 East Green Street Pasadena, CA 91106	1
Office of Naval Research New York Area Office 715 Broadway - 5th Floor New York, NY 10003	1

Naval Research Laboratory Technical Information Division Code 2627 Washington, D.C. 20380	6
Naval Ocean Systems Center Advanced Software Technology Division Code 5200 San Diego, CA 92152	1
Dr. A.L. Slafkosky Scientific Advisor Commandant of the Marine Corps (Code RD-1) Washington, D.C. 20380	1
Mr. E.H. Gleissner Naval Ship Research & Development Center. Computation & Mathematics Dept. Bethesda, MD 20084	1
Captain Grace M. Hopper NAICOM/MIS Planning Branch (OP-916D) Office of Chief of Naval Operations Washington, D.C. 20350	1
Mr. Kin B. Thompson NAVDAC 33 Washington Navy Yard Washington, D.C. 20374	1
Advanced Research Projects Agency Information Processing Techniques 1400 Wilson Boulevard Arlington, VA 22209	1