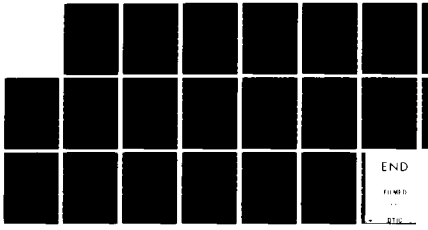
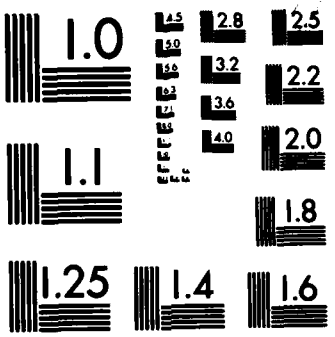
AD-A131 280 WHAT'S IMPORTANT ABOUT KNOWLEDGE REPRESENTATION(U) BOLT 1/1
BERANEK AND NEWMAN INC CAMBRIDGE MA W A WOODS JUL 83
BBN-5374 N00014-77-C-0371

UNCLASSIFIED F/G 6/4 NL





Bolt Beranek and Newman Inc.

(12)

Report No. 5374

What's Important about Knowledge Representation

Technical Report

W.A. Woods

July 1983

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. 5374	2. GOVT ACCESSION NO. AD-A131280	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) WHAT'S IMPORTANT ABOUT KNOWLEDGE REPRESENTATION		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER BBN Report No. 5374
7. AUTHOR(s) W. A. Woods		8. CONTRACT OR GRANT NUMBER(s) N00014-77-C-0371
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 10 Moulton St. Cambridge, MA 02238		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy Arlington, VA 22217		12. REPORT DATE July 1983 13. NUMBER OF PAGES 23
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)

Knowledge representation, semantic networks, KL-ONE, knowledge-based systems, rule-based systems.

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

This report discusses a number of issues that serve as research goals for the discovery of general principles of knowledge representation. It is concerned with the question of what constitutes a good representational system and a good set of representational primitives to deal with an open-ended range of knowledge domains. The discussion is illustrated with techniques and concepts evolved in the development of the knowledge representation system KL-ONE.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

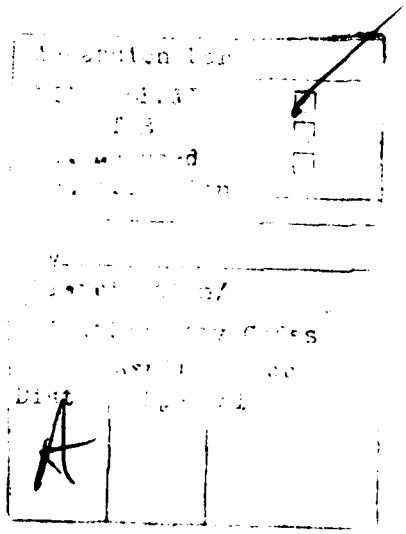
Report No. 5374

WHAT'S IMPORTANT ABOUT KNOWLEDGE REPRESENTATION

Technical Report

July 1983

Principal Investigator:
William A. Woods
(617) 497-3361



Prepared by:

Bolt Beranek and Newman Inc.
10 Moulton Street
Cambridge, Massachusetts 02238

Prepared for:

The Office of Naval Research

This research was supported by the Office of Naval Research under Contract No. N00014-77-C-0371. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Office of Naval Research or the U.S. Government.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

It is axiomatic in computer science that a good solution often depends on a good representation. For most artificial intelligence applications, the choice of representation is more difficult than usual, since the space of possibilities is substantially greater and the criteria with which to make choices are less clear. For representing the states of reasoning and states of knowledge of intelligent agents that can understand natural language or characterize perceptual data, the representation problem takes on extreme importance; the representational primitives and the system for their combination effectively limits what such systems can perceive, know, or understand.

In this paper, I will discuss a number of issues that serve as research goals for the discovery of general principles of knowledge representation. I am concerned with the question of what constitutes a good representational system and a good set of representational primitives to deal with an open-ended range of knowledge domains. By "representational primitives" here, I mean to include not just primitive concepts, but (more importantly) the primitive elements and operators out of which an open-ended range of learned concepts can be constructed. I will illustrate the discussion with techniques and concepts evolved in the development of the knowledge representation system KL-ONE.

The issues of interest will be a set of problems that arise

in attempting to construct intelligent computer programs that use knowledge to perform some task. Such problems include:

1. How to structure a representational system that will be able, in principle, to make all of the distinctions that may be important;
2. How to remain noncommittal about details that cannot be resolved;
3. How to recognize efficiently what knowledge is relevant to the situation in which the system finds itself;
4. How to acquire knowledge dynamically over the "lifetime" of the system; and
5. How to assimilate pieces of knowledge in the order in which they are encountered rather than requiring a specific order of presentation.

2. TWO ASPECTS OF KNOWLEDGE REPRESENTATION

In addressing issues of knowledge representation, there are two general aspects of the problem that need to be considered. The first has to do with the expressive power of the representation, i.e., what it can say. I will refer to this as expressive adequacy. (There are two components to expressive adequacy: (1) what distinctions a representation can make and (2) what distinctions it can leave unspecified in order to express partial knowledge.) The second general aspect of the knowledge representation problem concerns the actual shape and structure of the representation itself and the impact of this structure on the operations of the system. I will refer to this as notational efficacy. (Notational efficacy in turn breaks down into components such as computational efficiency for various kinds of inference, conciseness of representation, ease of modification, etc.)

It is important to distinguish expressive adequacy and notational efficacy, since there are a variety of arguments in this field that are exacerbated by failure to distinguish which issue is being addressed. For example, an argument that first-order predicate calculus should be used because it has a well-understood semantics attempts to partially address expressive adequacy, but does not explicitly mention the issue of notational efficacy. The argument could be taken to advocate use of the notations traditionally used by logicians (and in some

cases this may even be what is meant), but it is possible to invent many different notational systems all having a first-order-logic semantics, but having different behavior with respect to various components of notational efficacy. (For discussion of other aspects of the first-order-logic debate, see Israel [1983], this volume.)

In order to provide reasonable foundations for practical use of knowledge in reasoning, perception and learning, research in knowledge representation should seek notational conventions that simultaneously address issues of expressive adequacy and notational efficacy. What is required is a representational system that will be adequate for a comprehensive range of different kinds of inference and will provide computational advantages to inferences that must be performed often and rapidly. In the next section, I will argue that one class of inference that must be performed rapidly and efficiently is the characterization of one's current situation with respect to a taxonomically organized knowledge network.

3. THE ROLE OF A KNOWLEDGE NETWORK FOR AN INTELLIGENT MACHINE

In constructing an intelligent computer agent, a fundamental problem is analyzing a situation in order to determine what to do. For example, many expert systems are organized around a set of "production rules", a set of pattern-action rules that characterize the desired behavior of the system [5]. Such a system operates by determining at every step what rules are satisfied by the current state of the system and then acting upon that state by executing one of those rules. Conceptually this entails testing each of the system's rules against the current state, but as the number of rules becomes large, techniques are sought to avoid actually testing all of them.

My approach to the problem of determining what rules apply has been to assume that the pattern parts of all such rules are organized into a structured taxonomy of all the situations and objects about which the system knows anything. By a taxonomy, here, I refer to a collection of concepts linked together by a generality relation, so that concepts that are more general than a given concept are accessible from it. By a structured taxonomy I mean that the concept descriptions have internal structure that is available to the computer system so that (for example) the placement of concepts within the taxonomy can be computationally determined. Such a taxonomy has the characteristic that information can be stored at its most general level of applicability and indirectly accessed by more specific concepts, which are said to "inherit" that information.

If such a taxonomic structure is available, then the action parts of the system's rules can be attached to nodes in this structure as pieces of "advice" that apply to situations described by those rules. The task of determining the rules applicable to a given situation then consists of classifying the situation within this taxonomy and inheriting the advice. Thus, a principal role that a knowledge network can play for such a system is to serve as a "coat rack" upon which to hang various procedures or methods for the system to execute. Such a conceptual taxonomy can organize the pattern parts of a system's rules into an efficient structure that facilitates recognition. I will have more to say about this after presenting a brief introduction to one such system, KL-ONE, in the next two sections.

4. THE STRUCTURE OF CONCEPTS IN KL-ONE

In building up internal descriptions of situations, one needs to make use of concepts of objects, substances, times, places, events, conditions, predicates, functions, individuals, etc. Each such concept can be characterized as a configuration of attributes or parts, satisfying certain restrictions and standing in specified relationships to each other. This notion of concept is the basic element of the knowledge representation system KL-ONE [2].[1]

A concept node in KL-ONE consists of a set of roles (a generalization of the notions of attribute, part, constituent, feature, etc.) and a set of structural conditions expressing relationships among them. Concepts are linked to more general concepts by a relation called SUPERC. The more general concept in such a relationship is called the superconcept and is said to subsume the more specific subconcept. Some of a concept's roles

---

[1] KL-ONE is the result of the collaborative design of a number of researchers over an extended period of time, within which it is difficult to allocate credit to individual contributors. Principal developers have been Ron Brachman, Rusty Bobrow, Jim Schmolze, David Israel, and I, with contributions from Hector Levesque, Bill Mark, Tom Lipkis, and other people too numerous to mention. Within such a large group, there have been substantial differences in point of view as to what KL-ONE is or is attempting to be, and substantial evolution of those views over time. What I say here represents largely my own current view based on this experience, some of which is the result of substantial consensus with my colleagues and some of which is not.

and structural conditions are attached to it directly, while others are inherited indirectly from more general concepts.

The concepts and roles of KL-ONE are similar in structure to the general data-structure notions of record and field or the "frame"/"schema"/"unit" and "slot" of much AI terminology. However, there are several differences, such as the way subsumption is defined and used, the presence of structural conditions attached to a concept, explicit relationships between roles at different levels of generality, and the general intent of KL-ONE concepts to model the semantics and conceptual structure of an abstract space of concepts (rather than to play the role of a data structure in a computer implementation).

This last point may require some elaboration. I should point out that the goal of KL-ONE is not per se to produce a particular computer system, but rather to force the discovery and articulation of general principles of knowledge organization and structure. The issue of expressive adequacy is an important driving force in KL-ONE research. It leads to an emphasis on the semantics of the representation and its adequacy to make the kinds of subtle distinctions that can be made by people in conceptualizing complex ideas. (See [6] for a discussion of the importance of the issue of semantics of a semantic network.) Thus, the KL-ONE effort has had much more of a spirit of applied philosophical investigation than much other work in knowledge representation and data structures.

5. AN EXAMPLE OF A CONCEPTUAL TAXONOMY

Space does not permit a complete exposition of KL-ONE here. However, the kind of taxonomic structure it provides is illustrated in Figure 1.

In this figure, concepts are represented by ellipses and roles by circled squares. At the top of the figure is a high-level concept of Activity. This concept has roles for Time, Place, and Participants, which are inherited by all concepts below it. Immediately below Activity is the concept for a Purposive Activity, which differentiates (DIFFS) the general role for Participants into an Agent (which is the participant that has the purpose) and Other Participants. Purposive Activity introduces a new role called Goal to represent the purpose of the activity.

Below this is a fairly specific (but still generic) concept for Driving to Work. This concept modifies the Goal of Purposive Activity by adding a value restriction (V/R) Getting to Work, indicating that whatever fills the Goal role must be an instance of Getting to Work. It also introduces a new role, called Destination, whose value restriction is a Place of Work. A structural condition (not shown) attached to the concept would specify how the Place of Work related to the Getting to Work goal (i.e., it is the Destination of the Getting to Work goal). Driving to work in Massachusetts is, in turn, a specialization of Driving to Work, whose Destination is restricted (MODS) to be a

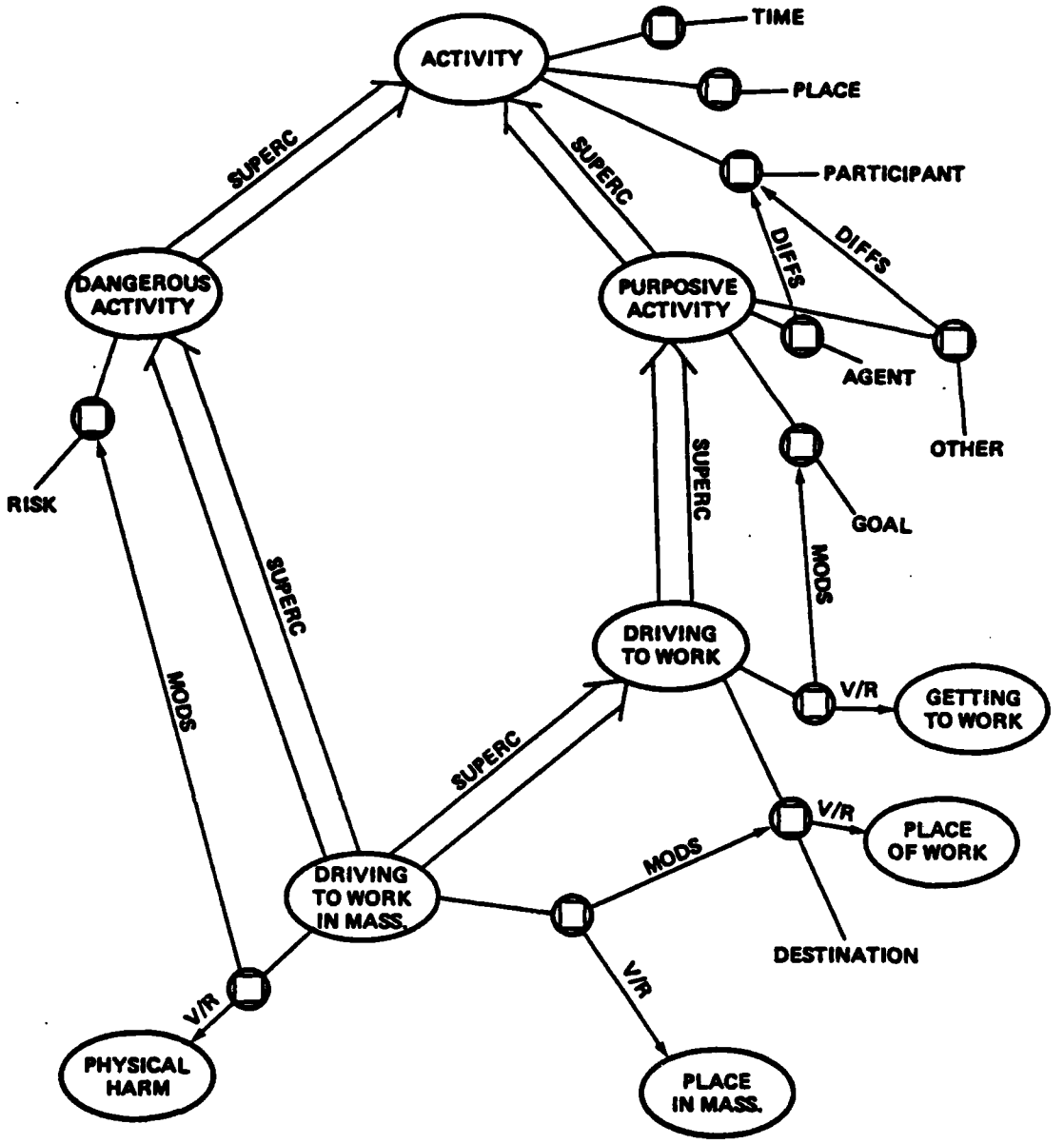


FIG. 1. AN EXAMPLE OF A KL-ONE NETWORK.

Place in Massachusetts. It is also a specialization of a Dangerous Activity whose Risk is Physical Harm.

This figure illustrates the kind of taxonomy that one would expect to have in an intelligent computer agent. It includes both very high level abstractions and quite specific concepts. Moreover, there is always room for the insertion of new levels of abstraction in between existing ones. In fact, there is a well-defined classification procedure implemented in the KL-ONE system that can automatically place a new description into such a taxonomy linked by SUPERC connections to the concepts which most specifically subsume it and those which it in turn subsumes.

6. THE NEED FOR TAXONOMIC ORGANIZATION

Having now introduced some KL-ONE terminology and a rough idea of what a KL-ONE network looks like, let us return to the problem of recognizing what rules apply to a situation and see what such a taxonomy can do for us. In most expert system applications, a task description will often satisfy several rules simultaneously, no one of which will account for all of the task nor supplant the relevance of the others. For example, adding an object to a display is simultaneously an example of changing a display and of displaying an object. Advice (i.e., the action parts of the rules) associated with both activities must be considered. Moreover, one situation description may subsume another (more specific) description and their advice may either supplement or contradict each other. Thus, conventions are required to determine which advice takes precedence when conflicts arise.

For independent rules in a classical production rule system, such conflicts are only discovered when an instance of a conflicting situation occurs as input. When using a taxonomic classification structure, however, the subsumption of the conditions of one rule by another can be discovered when the rule is assimilated into the taxonomy, at which time the person entering the rule can address the question of how the two rules should interact. The advice associated with the more specific rule can then explicitly include the information to override or supplement the more general rule.

This view of assimilating rules into a taxonomic knowledge structure not only facilitates the discovery of interactions at input time (one element of notational efficacy), but also promotes a compactness in the specification of the rules themselves. By relying on the fact that concepts inherit information from more general concepts, one can usually create the concept for the pattern part of a new rule by merely adding a minor restriction to an existing concept. In KL-ONE, when one wants to create a situation description that is more specific than a given one, it is only necessary to mention those attributes that are being modified or added; one does not have to copy all of the attributes of the general situation. Aside from conserving memory storage, this also facilitates updating and maintaining the consistency of the data base by avoiding the creation of duplicate copies of information that may then need to be independently modified, and could accidentally be modified inconsistently. This is yet another element of notational efficacy.

The ability to assimilate new descriptions into an existing taxonomy at any level permits an evolutionary system design that achieves the same standards of rigor as a "top-down design" without requiring concepts to be defined in a predetermined order. For most applications, even if one could get the initial design carefully laid out in a rigorous top-down mode, subsequent changes (e.g., required changes in accounting policies induced by new tax laws) will require an ability to modify a system in more

flexible ways. A system's taxonomy of recognizable situations should be viewed as an evolving knowledge structure that continues to be refined and developed throughout the lifetime of a system, just as it is for human beings.

7. PARSING SITUATIONS

In addition to the advantages discussed above, the use of a taxonomic structure can have considerable advantages for the process of recognizing that some of the elements currently being perceived constitute an instance of a known situation. Roughly, this process consists of discovering that those elements can be interpreted as filling roles in situation descriptions known to the system. However, it is not usually sufficient to characterize a situation as a single instance of an existing situation description. In general, a situation description must be a composite structured object, various subparts of which will be instances of other concepts assembled together in ways that are formally permitted, in much the same way that the description of a sentence is put together from instances of various kinds of phrases.

Thus, the process of recognizing a situation is somewhat similar to the process of parsing a sentence, although considerably more complex, due to a more open-ended set of possible relationships among the "constituents" of a situation. Whereas sentence grammars deal mainly with adjacency of phrases, the relationships among constituents of a situation may be arbitrary, including: events preceding one another in time; people, places, and physical objects in various spatial relationships with each other; people in physical or legal possession of objects; people in relationships of authority to other people; and people having certain goals or objectives.

One technique for improving the efficiency of situation recognition is to use what I have called a "factored" knowledge structure [7]. In such a structure, the common parts of different rules are merged so that the process of testing them is done only once. With such structures, one can effectively test a large set of rules without considering each rule individually. The kinds of taxonomic structures embodied in KL-ONE can provide such a factored representation for the parsing of situations. This can be done by using chains of links from elements of the situation to roles of higher level concepts in which they can participate, together with generalizations (and extensions) of the kinds of algorithms used to parse sentences, in order to determine the most specific concepts that subsume the input situation.

The suitability of a representation to support algorithms of this sort is an important component of notational efficacy. A version of this kind of technique using KL-ONE has been successfully applied in the PSI-KLONE system, an ATN transducer coupled with a KL-ONE taxonomy organizing the semantic interpretation rules of a natural language understanding system [1].

8. EXPRESSIVE ADEQUACY

However efficient a representation may be for some purposes, it is all for naught if it can't express the distinctions that are necessary. In seeking a representation in which it will be possible to represent any distinction that may become important, one must avoid choosing a set of primitives that washes out subtle distinctions such as those between "walk", "run", "amble", "drive", and "fly". On the other hand one should not miss the advantages of the commonality between these and the general concept "move". A structured inheritance network such as KL-ONE permits one to gain the benefits of both. It is always possible to introduce new distinctions as they become important by the refinement or modification of existing concepts, and it is always possible to introduce more general concepts that abstract away detail of more specific ones. One aspect of the use of KL-ONE for such problems is illustrated in the following.

A common problem in axiomatizing a domain in the traditional predicate calculus notations is choosing the set of predicates and deciding what arguments they will take. Inevitably these decisions leave out distinctions that for another purpose might be important, such as time variables, situation variables, provisions for manner adverbial modification, intermediate steps and agents, etc. Incorporating revisions to these decisions in a complex system would amount to redoing the axiomatization. One of the goals of KL-ONE, toward which some progress has been made,

is to provide a terminological component for such axiomatizations (i.e., KL-ONE concepts provide the predicate inventory for the axiomatization), so that, for example, the fact that activities have a time role (see Figure 1) can be virtually ignored in expressing an axiom in which time does not figure prominently, and yet remain present implicitly (or be added later) and play its role when a situation is encountered in which it becomes important.

9. A VIEW OF THE FUTURE

Expressive adequacy is a minimal requirement for a knowledge representation system; eventually, one wants a framework in which the assimilation of arbitrary new information is not only possible, but is in some sense natural. For example, one would want small changes in knowledge to require small changes in the knowledge base, so that processes of learning (or even incremental debugging) can be expected to eventually converge. Moreover, there must be operators for making fine adjustments as one gets close to the correct state. Thus, another element of notational efficacy includes some analog of the mathematical property of compactness -- i.e., there should be points in the space of possible states of knowledge that are arbitrarily close to the state of knowledge one wants to achieve.

I have argued that taxonomic classification structures can provide advances in both expressive adequacy and notational efficacy for intelligent systems. I believe that such techniques will eventually have widespread applicability in all of computer science. The emphasis on the expressive adequacy of representation, rather than (primarily) the computational efficiency of data structures, holds the promise of developing a general methodology of representation that will transcend different applications and different implementation techniques. Ultimately, this should lead to a way of specifying computational behavior in terms of high-level conceptual operators that match

the conceptual structures of the human programmer, factoring out for separate consideration (or even automatic compilation) the issues of implementational efficiency.

The beginnings of such a trend are already occurring in the form of increased emphasis on abstract data types and "object-oriented" programming [4]. The next logical step is the generalization of the notion of abstract data types to the level of abstraction, inheritance, and expressive adequacy present in a sophisticated knowledge representation system. This could produce a new style of programming which Goodwin [3] has labeled "taxonomic programming." This style of programming can have enormous advantages for flexibility, extensibility, and maintainability, as well as for documentation, user education, error reduction, and software productivity. Moreover, such representations can make it possible to combine independently developed systems to produce integrated systems that are more powerful than the mere union of their parts.

Acknowledgments

Research described here was supported in part by the Office of Naval Research under Contract No. N00014-77-C-0371, and by Defense Advanced Research Projects Agency under Contract No. N00014-77-C-0378.

REFERENCES

[1] Bobrow, R.J. and Webber, B.L.
Knowledge Representation for Syntactic/Semantic Processing.
In Proceedings of The First Annual National Conference on Artificial Intelligence, pages 316-323. American Association for Artificial Intelligence, 1980.

[2] Brachman, R.J. and Schmolze, J.
An Overview of the KL-ONE Knowledge Representation System.
Cognitive Science, 1984.

[3] Goodwin, James W.
Taxonomic Programming with Klone.
Technical Report, Informatics Laboratory, Linkoeping University, Sweden, 1979.

[4] Robson, D.
Object-Oriented Software Systems.
Byte 6(8):74-86, August, 1981.

[5] Waterman, D.A. and Hayes-Roth, F.
An Overview of Pattern-Directed Inference Systems.
In D.A. Waterman and F. Hayes-Roth (editors), Pattern-Directed Inference Systems, pages 3-22. Academic Press, Inc., New York, NY, 1978.

[6] Woods, W.A.
What's in a link? Foundations for semantic networks.
In D.G. Bobrow and A. Collins (editors), Representation and Understanding: Studies in Cognitive Science, pages 35-82. Academic Press, New York, 1975.

[7] Woods, W.A.
Cascaded ATN Grammars.
Amer. J. Computational Linguistics 6(1):1-15, Jan.-Mar., 1980.

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
Naval Research Laboratory Technical Information Division Code 2627 Washington, D.C. 20380	6

cont'd.

Naval Ocean Systems Center 1
Advanced Software Technology Division
Code 5200
San Diego, CA 92152

Dr. A. L. Slafkosky 1
Scientific Advisor
Commandant of the Marine Corps
(Code RD-1)
Washington, D.C. 20380

Mr. E. H. Gleissner 1
Naval Ship Research & Development Ctr.
Computation & Mathematics Dept.
Bethesda, MD 20084

Capt. Grace M. Hopper, USNR 1
Naval Data Automation Command
Code 00H
Washington Navy Yard
Washington, D.C. 20374

Mr. Paul M. Robinson, Jr. 1
NAVDAC 33
Washington Navy Yard
Washington, D.C. 20374

Advanced Research Projects Agency 1
Information Processing Techniques
1400 Wilson Boulevard
Arlington, VA 22209

Capt. Richard L. Martin, USN 1
507 Breezy Point Crescent
Norfolk, VA 23511

Director, National Security Agency 1
Attn: R54, Mr. Page
Fort G.G. Meade, MD 20755

Director, National Security Agency 1
Attn: R54, Mr. Glick
Fort G.G. Meade, MD 20755

Major James R. Kreer 1
Chief, Information Sciences
Dept. of the Air Force
Air Force Office of Scientific Research
European Office of Aerospace
Research & Development
Box 14
FPO New York 09510

cont'd.

Mr. Fred M. Griffee
Technical Advisor C3 Division
Marine Corps Development
& Education Command
Quantico, VA 22134 1