Selecting An Object Oriented Paradigm[1]

Jacob L. Cybulski
Department of Computer Science
Royal Melbourne Institute of Technology

Abstract

This paper summarizes some of the results of a project in evaluation and selection of an object-oriented paradigm of representation and programming. A number of different object-oriented models were analysed for their capability of capturing declarative and procedural knowledge, for the economy of their representation, for the variety and flexibility of knowledge manipulation tools, and finally for the models' ability to efficiently order and subsequently select knowledge units for processing. An assessment is made of the programmer's effort to apply each of the methodologies to the area of an knowledge representation and manipulation, exemplified by an artificial domain of the robot construction and its action planning.

Paper Plan

---

[1] Note portions of this article are based on an unpublished report developed by the author at Knowledge Systems Laboratory at Royal Melbourne Institute of Technology, Melbourne, Victoria, March 1987.

1 Introduction

One of the most important tasks in knowledge engineering is to successfully match an application at hand with one of many knowledge representation and processing tools. The tools selected will usually fit into one of the three programming models :- the *functional*, the *logical* and the *object-oriented*. Each of the methodologies evolved into a vast and complex domain of techniques, methods and tools tailored to specific applications, having their own idiosyncrasies, overlapping, converging and complementing each other. The selection task is difficult, even after the methodology itself is already chosen.

The functional and logical approach to knowledge representation and processing is relatively well researched and documented, both have good theoretical background, and both have well publicized algorithms, methods and techniques. The object-oriented approach to programming is quite new and in its theoretical and developmental infancy. Thus, there is still a number of questions pending in front of every knowledge engineer inclined to use an object-oriented paradigm:

- What is an object-oriented model of knowledge representation?
- What are the necessary components of an object-oriented system?
- What are the criteria for the selection of an object-oriented system?
- How difficult is to implement an object-oriented tool?

The author of this paper was confronted with these questions during a development of an object-oriented system CONTEXTUS applied to the problems of a robot action planning (CYBULSKI 1987). The questions were subsequently answered and this paper summarizes some of the findings.

2. What is an object-oriented model of knowledge representation?

Object-orientation will be defined as a modularization of representational elements, both declarative and procedural, where modules, whether synchronous or asynchronous, are independent but coupled together via communication channels. This definition allows quite a number of knowledge representation methodologies to be considered "object-oriented", including frames, actors, conceptual structures, blackboards, some production systems, activation and inference networks, neural networks, and others (refer to WINSTON 1984 for a general overview of different representation techniques).

The analysis of knowledge representation models considered in this paper was inspired by the availability of and experience with the specific Artificial Intelligence tools: PEARL representing *frame systems* (DEERING, FALETTI AND WILENSKY 1982, FALETTI AND WILENSKY 1982), NEOACTORS representing *message passing systems* (CYBULSKI 1987), MICROEXPERT representing *activation networks* (COX AND BROUGHTON 1982), finally a few low level tools such as LISP (FRANZLISP - WILENSKY 1984; SCHEME - ABELSON, SUSSMAN AND SUSSMAN 1985) and PROLOG (CLOCKSIN AND MELLISH 1981); discussion of other methodologies and techniques is based on broad literature survey to include complicated hybrid architectures combining the elements of rule and object-oriented representations (e.g. LOOPS or KEE - KUNZ, KEHLER AND WILLIAMS 1984).

A more general treatment of object-oriented systems (as defined here) may be found in SHRIVER AND WEGNER 1987 (actors and arames), SOWA 1984 (conceptual structures) and RUMELHART AND MCCLELLAND 1986 (activation networks); other references can be found in the body of this paper.

3. What are the necessary components of an object-oriented system?

The CONTEXTUS system was designed as a general purpose object-oriented system, it was applied however to the problems of action planning by robots. Hence, the system must have had the necessary model components to suit the application. It was important to identify a number of universal, primitive concepts to describe the robot world (mapping), complex robot systems had to be defined in terms of inter-related parts and functions (organization), the parts needed to be grouped and classified according to their characteristics (clustering), finally the action planning required dynamic prioritization of goals, plans and actions (selection). Capability to *map*, *organize*, *cluster* and *select declarative* and *procedural* knowledge in the robot realm were the ultimate goal of every tool and methodology considered, hence, let us explore the four areas of knowledge representation as applicable to our robot example.

Mapping. Many of the researchers into Psychology and Cognitive Science find evidence that human memories, and thus knowledge, could be mapped into a small set of representational universals. Schank for instance proposed 11 primitive concepts and 16 primitive inferences (SCHANK 1975), others believe in a much larger set of primitive semantic units (e.g. 60 in WILKS 1973). Even though most of the work on representational universals were done in the area of language understanding, the primitives could be drawn from any application domain and they could be of any conceptual complexity; e.g. cognitive processes, conceptualizations, plans and goals can all be reduced to simple, primitive structures and mechanisms (SCHANK AND ABELSON 1977). From the computational point of view, a small set of primitives

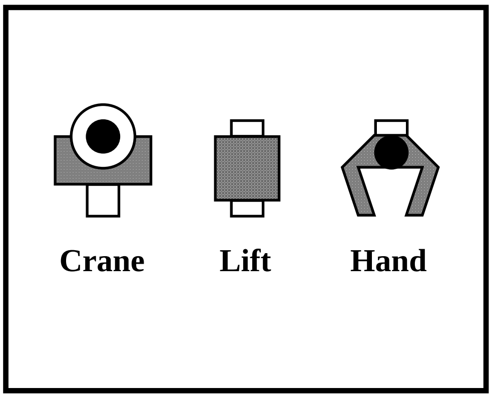

Fig. 1 - Knowledge primitives

allows a more complete definition of all the valid and invalid relationships that may occur between pairs of primitive concepts, it is also possible to provide details of transformations reducing complex concepts into the simple ones.

Thus, considering an area of a robot construction and its action planning a set of primitives may include all basic robot parts: cranes, lifts and hands. The primitive functions characterize each of the parts and the necessary causal relationships between their behaviours are defined.

Organization. There are many models of knowledge organization. By far, *semantic network* theory is the most popular model of declarative knowledge. It was developed by Quillian (1968) and later extended by a number of AI researchers leading to the development of natural language applications (SIMMONS 1973, SHANK 1975), numerous theories of memory such as active semantic networks (RUMELHART AND NORMAN 1973) and frame systems (MINSKY 1975), logics systems, etc. (PARTITIONED SEMANTIC NETWORKS - HENDRIX 1978, CONCEPTUAL GRAPHS - SOWA 1984; for general discussion of semantic networks see WOODS 1975 and FINDLER 1978A).

Psychological experiments support the semantic network hypothesis that human memory may be viewed as a network consisting of a huge number of inter-related concepts. The basic representation components are *concepts* (representing objects, situations, persons) and *relationships* (representing time, space or conceptual dependencies, object properties, acts and doings). In the computer terms any real-life object or a situation may be described by an appropriate network of connections manipulated by the standard graph-theoretical algorithms.

For instance a simple robot arm consisting of three primitive parts (a hand, a lift and a crane - figures 1 and 2) may be represented by three concepts linked by four relations denoting the parts spatial relationship

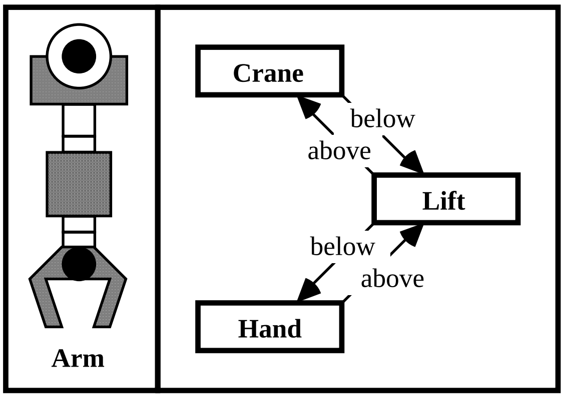

Fig. 2 - Representation of objects

(above and below). A representation of a non-arm object consisting of the same parts will have different topological properties, thus distinguishing it from the standard robot arm.

As the semantic nets capture the declarative knowledge in a rigid and formal manner, few of the artificial intelligence techniques deal with the procedural aspects of representation in an organized fashion. *Message passing*, firstly introduced by Hewitt in his *actor system* (PLASMA - HEWITT, BISHOP AND STEIGER 1973, ACTOR - HEWITT 1977), is one of the very few methodologies that have very good formal background. The methodology is, in many respects, similar to those of other AI formalisms, that of *production systems* (DAVIES AND KING 1977, WINSTON 1984 P 201-203) and that of *blackboard architectures* (NII 1986A, 1986B). The philosophy of the actor, production and blackboard systems is based on a common concept of message passing through the network of *actors* or processors, called "actors" in actor systems, "productions" in production systems, or "knowledge sources" in blackboard architecture. Every actor provides some service to other actors of the system by invoking its local procedure, also known as an actor’s *method*. A *message*, being a request for service together with some additional information, is deposited in a *buffer* known to a group of actors; the message depository is known as a "mail queue" in an actor system, "context" in production systems, and "blackboard" in blackboard systems. The actors inspect all the buffers they have access to and serve all the requests from the buffer they are able to, the service may involve sending new messages to some other buffers in the system. Actors usually specialize in a specific domain of service they could provide. In some systems, thus, an actor may be able to perform only one specific action (as in SOWA 1984), in others it may be allowed to change its behaviour as a result of the action performed (serialized behaviour, as in AGHA 1986), or else an actor may be able to perform a wide range of related activities (as in LOOPS - STEFIK AND BOBROW 1986). It should also be noted that most frame representation systems use the semantic structures in combination with a message passing methodology for knowledge representation and manipulation (e.g. SMALLTALK - GOLDBERG AND ROBSON 1983, or COMMONLOOPS - BOBROW ET AL 1985).

Message passing systems are especially suited for procedural-intensive applications. The robot action description is certainly an example of such an application. The robot arm parts act asynchronously and may communicate only via the specific channels (buffers). Figure 4 shows the different types of messages sent between an arm’s crane, its lift and the hand while moving an object on the floor.

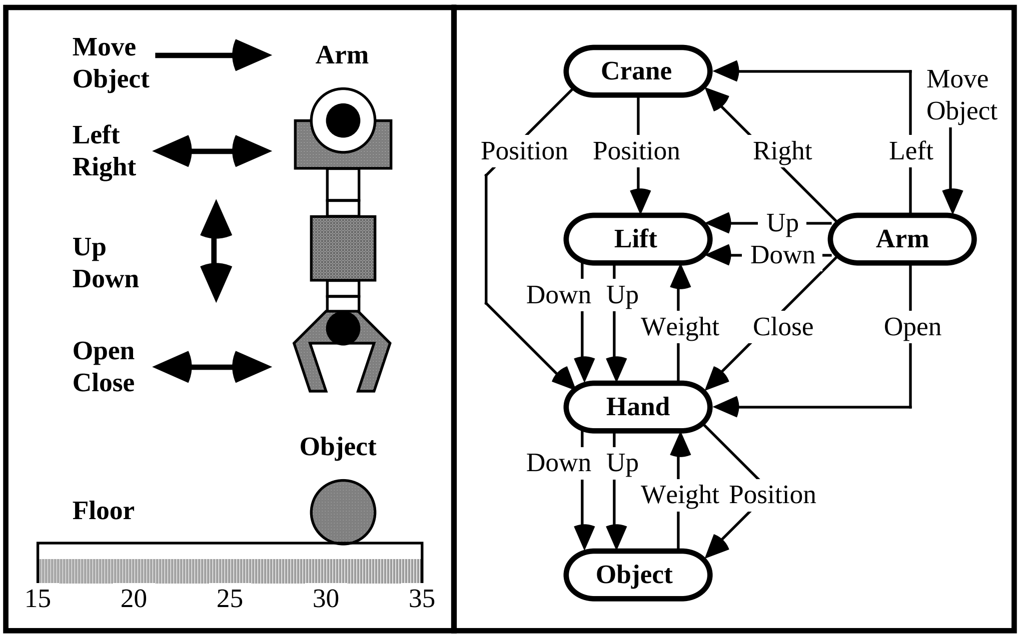

Figure 3 - A message passing system

Clustering. The concepts of "semantic networks" and "message passing" may be greatly improved by an introduction of aggregate descriptions for concepts, relationships and procedures as inspired by psychological evidence that human thoughts are organized into schematic structures and processes. The notion of memory schemata was first put to the AI community by Minsky (1975), who summarized all preceding attempts to represent knowledge in semantic, taxonomic and aggregate forms and proposed a new knowledge representation formalism known as a *frame representation system*.

The schematic organization of memory imposes a taxonomy of description types where the references to real-life objects are treated as *instances* of abstract descriptions known as their *prototypes*. The prototypes may also be structured hierarchically, in which case the more general descriptions are referred to as *super-classes*, the more specialized ones as *sub-classes*. The main advantage of such organization is economy of representation for all objects stemmed from the same prototype share their characteristics. The frame system's ability to infer object properties from their prototype description is known as *inheritance*.

For instance, all possible robot parts, may be classified depending on their shape, colour, function or the type of mounting. The parts parts may be classified as being either cranes, lifts or hands, each of them may be further sub-classified on the type of mounting in the robot arm (figure 3). All robot parts share the property of metallic colour (inheritable from "Robot Part"), all lifts' function is to perform vertical movements (inheritable from "Lift"), and all hands perform grasping actions (from "Hand").

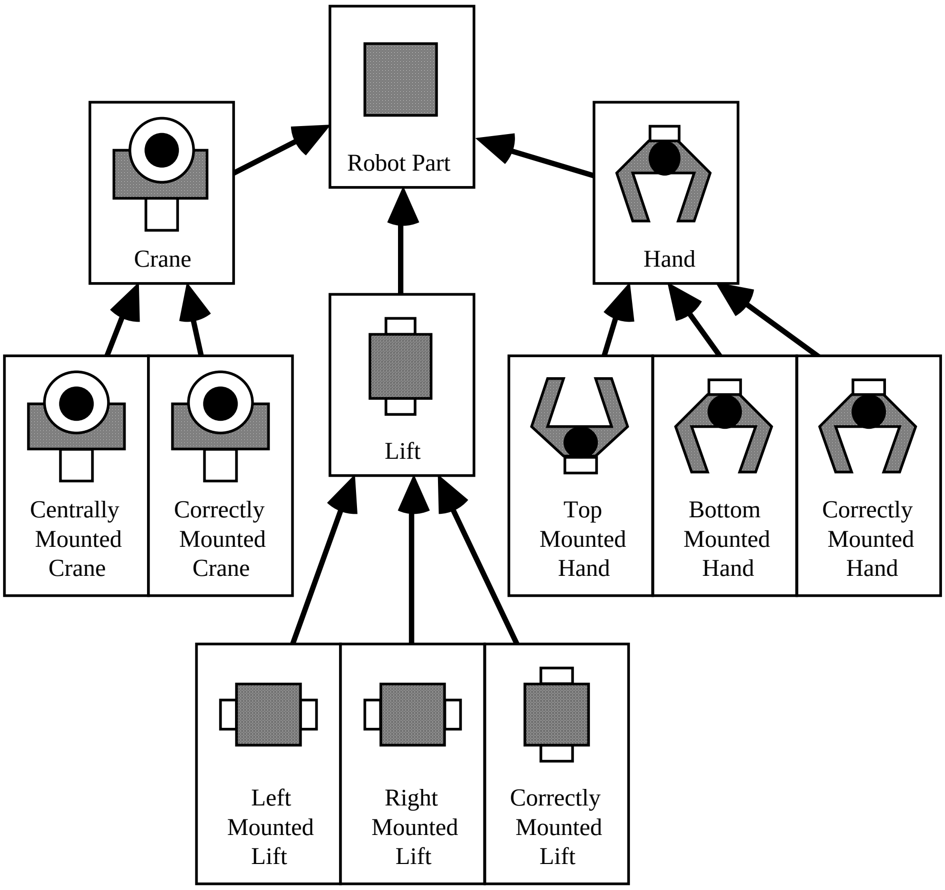

Figure 4 - Concept taxonomy

A number of AI researchers have followed Minsky's idea of frame representation and a number of frame-like organizations became common in knowledge manipulation systems, for instance in FRL (ROBERTS AND GOLDSTEIN 1977), UNITS (STEFIK 1979, 1980), TAXIS (MYLOPOULOS, BERNSTEIN AND WONG 1980), PEARL (DEERING, FALETTI AND WILENSKY 1982, FALETTI AND WILENSKY 1982), SMALLTALK (GOLDBERG AND ROBSON 1983), FLAVORS (ALLEN, TRIGG AND WOOD 1985), KL-ONE (BRACHMAN AND SCHMOLZE 1985), and KRYPTON (BRACHMAN, FIKES AND LEVESQUE 1983). Frames have been used in a number of different applications, e.g. in flight booking systems (GUS - BOBROW ET AL 1977), scheduling (NUDGE - GOLDSTEIN AND ROBERTS 1979), language understanding (SCHANK AND RIESBECK 1981), used as a new programming methodology (object oriented programming - BYTE 1986), or modelling human *conceptual structures* which usually possess incomparable capability to combine both declarative and procedural aspects of knowledge, e.g. semantic templates (WILKS 1973), scripts (SCHANK 1975), memory partitions (HENDRIX 1978), plans (SCHANK AND

ABELSON 1977), story schemata (RUMELHART 1975), conceptual graphs (SCHANK 1975, SOWA 1984, FARGUES ET AL 1986) and scenarios (MYLOPOULOS ET AL 1976).

It should be noted that taxonomic organization of knowledge could be useful in some of the semantic net applications, but that careless formalizations of taxonomic relations may lead to serious epistemological problems (BRACHMAN 1983, 1985). Further discussion of frame systems may be found in STEFIK AND BOBROW 1986, the formalization of frame properties, in HAYES 1979.

Selection. Because of knowledge complexity, because of its fuzziness and ambiguity, every knowledge based system must have appropriate mechanisms for efficient selection and retrieval of its knowledge structures. Since we selected a message passing system as means of communication between the procedural units of knowlege, we looked for a set of knowledge selection techniques that could enhance the selectivity of retrieval processes. We turned to the *activation networks* which could be viewed as a special case of a message passing system where the information flown between the actors is their activation charge, and the only action that may be taken is redistribution of activation to the inter-connected actors after reaching an activation threshold. The links between actors may be of *activation* (increase the activation of neighbouring actors) or *inhibition* (decrease the activation) type. A number of activation models are based on neurophysiological theories of the brain and its processing (MCCULLOCH AND PITT 1943, HEBB 1949), the theory of perceptrons (ROSENBLATT 1958, MINSKY AND PAPERT 1969), K-Lines and Societies of Mind (MINSKY 1977, 1980, 1986), and probabilistic inference networks (COX AND BROUGHTON 1982).

Knowledge in a true activation networks is stored as different activation states of all the network cells. Such an approach to information storage is similar to the concept of associative memory, or that of probabilistic inference networks (as in MICROEXPERT - COX AND BROUGHTON 1982, AL/X - REITER 1981, PROSPECTOR - DUDA ET AL 1978 or MYCIN - SHORTLIFFE 1976). Activation networks do not compute results, for the results may not flow in the net, instead the network settles into the state of a problem solution, i.e. its activation flow becomes stable. The low level of representation, very fine grain of knowledge units, high density of semi-random network of cell connection, and numerous theories of the rules governing the activation spreading (COLLINS AND LOFTUS 1975) usually requires very complex knowledge acquisition algorithms, all of which make the knowledge model unsuitable for direct programmer's manipulation (see HINTON 1985 or MICHALSKY AND CHILAUSKY 1981 for explanation of learning techniques in activation networks).

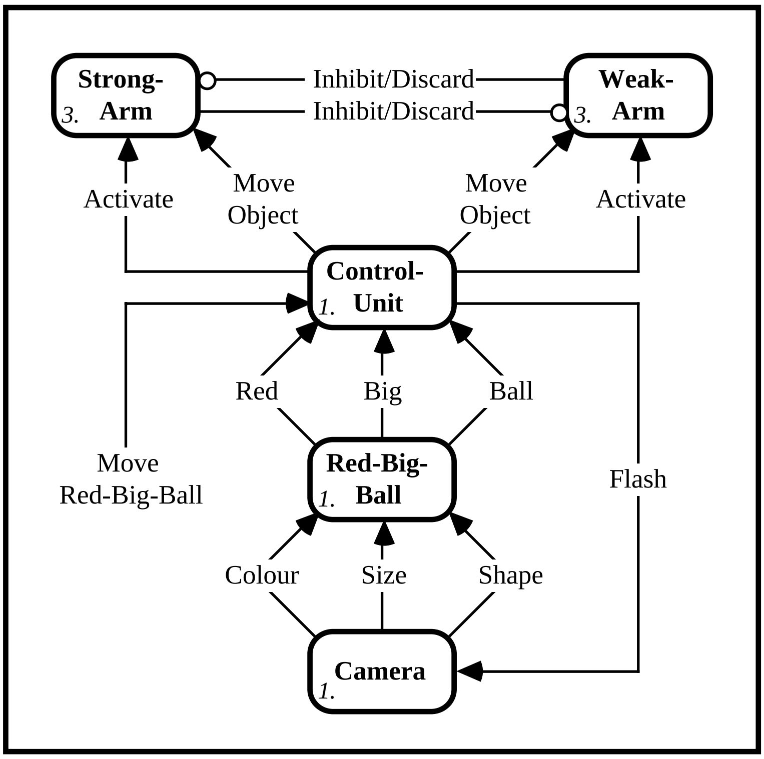

Figure 5 - Selection problem in activation networks

An application of a simple activation network principle may be illustrated by a situation of a multi-arm or multi-robot system where the suitability of the robot arms to perform a specific task is determined by the observation only (object colour, shape and size - figure 5). Each positive property identification (via a camera) will fire an activation charge to the more suitable arm (suitability based on previous experience) until its charge surpasseses a predefined threshold in which case the arm will fire thus inhibiting the other(s) from performing the action.

It should be stressed that an activation network organization is very different from a semantic model of memory. Some attempts have been made, however, to bridge the gap between semantic organizations and activation structures (see FELDMAN 1985). Others look at the possible combination of the neural model of knowledge representation with other semantically-oriented methodologies, proving such an approach to be highly successful (as in language parsing, combining activation nets and linguistic structures, POLLACK AND WALTZ 1984 also SMALL, COTTRELL AND SHASTRI 1982), such high level type of activation networks are especially well suited for ordering, sequencing and preemption of parallel, concurrent processes (CYBULSKI 1987).

4 Criteria for the selection of an object-oriented system?

Previous sections of this paper described a number of different models of knowledge representation (semantic networks, frame and actor systems, conceptual structures and activation networks), related to the knowledge manipulation tools available to the author (PEARL, NEOACTORS and MICROEXPERT). Each of the models addresses only some of the issues of object-orientation, and each of the tools considered would have to be further developed to cover all the necessary aspects to be fully applicable to the robot world.

This section looks back at all the systems discussed, performs a comparative analysis of the tools available, critically assesses the tools' suitability to object-oriented modelling, and selects the best combination of tools' features for the development of a system fully supporting Object Oriented Programming (OOPS). Apart from the tools mentioned we will also consider representational powers of LISP and PROLOG representing functional and logical tools respectively.

The assessment of each tool will be based on two representational categories, the declarative and procedural, for each of the areas of OOPS processing. Therefore, we will attempt to evaluate the following aspects of the tools:

mapping extensiveness and semantic simplicity of the set of procedural and declarative primitives;

organization the complexity and the cost of organizing semantic elements from the system's primitives (concepts and relationships);

clustering ease of grouping the semantic elements into clusters (more complex memory units and taxonomic classes);

selection selectivity of the system's processing mechanisms.

The rating for each category will be given in a scale - poor, bad, acceptable, good and excellent - for both the declarative and procedural aspects.

Each tool will then be assessed in terms of its usability for the implementation of the full OOPS model, and the tool's worst and best features. We will also look at the predicted performance of the resulting OOPS model, after it is constructed according to the guide-lines for the implementation of the OOPS elements; here we will look at the model complexity and expressiveness to predict future development efforts, and at the estimated computational complexity of the system, and hence its run time efficiency. The implementation and development effort, and the run time efficiency will be rated on the scale: low, medium and high.

5 How difficult is to implement an object-oriented tool?

Figure 6 abridges all the tools' characteristics and their assessment. The figure clearly shows SCHEME, PROLOG and MICROEXPERT to be the most primitive tools. Conceptual graphs to be the most elaborate. PEARL and NEOACTORS need extra effort put into their development to become useful tools.

SCHEME's (and LISP's) primitives are counter-intuitive and at a conceptually low level; the cost of knowledge organization with the use of symbols, a-lists and p-lists would be relatively high, also the implementation of fully independent parallel processing in knowledge units would have to be co-routined and be based on function prioritization.

PROLOG primitives are of a much higher level than LISP's, and they could be readily used in the creation of semantic networks; knowledge clustering would, however, require an imposition of an inference meta level (theories-like).

PEARL, similarly to other simple frame systems, has an immense capability to structure declarative knowledge; the procedural representations are quite elementary, though flexible (using normal lambda expressions). Frames are natural knowledge clusters. Their processing is dependent on demons and thus is a "side-effect" driven.

NEOACTORS specializes in the representation of procedural knowledge; its declarative powers are rather poor.[2] Actor systems are quite flexible in organizing different knowledge processing mechanisms.

MICROEXPERT has generally poor ratings; the main reason for this is the tool's inability to represent either declarative or procedural knowledge directly. Neural systems have immense hidden powers; they need, however, very strong and sophisticated knowledge acquisition mechanisms, then their knowledge selection via probability or activation could prove immeasurably useful.

---

2 Most of the actors-like system combine the power of frames and actors together. It is also the case with NEOACTORS.

Tool	Mapping	Organization	Clustering	Selection	Rating
Scheme	proc: accept. decl: accept.	proc: poor. decl: poor.	proc: accept. decl: accept.	proc: poor. decl: accept.	bad.
Functional Models.	symbols, values, lambda exps, functions, lists, a-lists, p-lists.	concepts - symbols; relationships - a-lists, p-lists; inferences - functions.	either lambda closures; or a-lists, p-lists.	filters - closures; amplification - process agenda.	implementation effort: medium. development effort: high. run time efficiency: high. worst feature: low level of expression. best features: flexibility, functional expression.
Prolog	proc: good. decl: good.	proc: accept. decl: accept.	proc: bad. decl: bad.	proc: bad. decl: bad.	bad.
Logic Systems.	values, variables, facts, rules.	concepts - symbols, variables; relationships - facts; inferences - rules;	either theories; or multiple databases.	filters - assertions, contexts, theories; amplification - meta-rules.	implementation effort: medium. development effort: medium. run time efficiency: low. worst feature: low level of expression. best features: flexibility, declarative expression.
Pearl	proc: bad. decl: good.	proc: bad. decl: good.	proc: good. decl: excellent.	proc: bad. decl: good.	good.
Frame Systems.	primitives, frames, slots, proto hierarchy, demons, servants.	concepts - frames; relationships - frames; inferences - demons, servants.	frames.	filters - perspectives, contexts, trans-frames; also - multiple inheritance.	implementation effort: medium. development effort: medium. run time efficiency: medium. worst feature: procedural representation. best features: structurality, declarative expression.
Neo-Actors	proc: good. decl: bad.	proc: good. decl: bad.	proc: excellent. decl: bad.	proc: good. decl: good.	accept.
Actor Systems, Production Systems, Blackboards.	actors, messages, channels, buffers.	concepts - actors, buffers; relationships - channels; inferences - actors.	actors, buffers.	filters - message, passing; amplification - buffer loads.	implementation effort: medium. development effort: medium. run time efficiency: medium. worst feature: declarative representation. best features: modularity, procedural expression.
Conceptual Graphs	proc: excellent. decl: excellent.	proc: excellent. decl: excellent.	proc: excellent. decl: excellent.	proc: good. decl: good.	excellent.
Scripts, Semantic Partitioned Networks, Scenarios.	primitives, concepts, relations, inferences.	as is.	clusters.	filters - plans, vistas, schemata; amplification - pattern matching.	implementation effort: high.. development effort: low. run time efficiency: low. worst feature: run time efficiency but may be compiled. best features: structurality, modularity, integration.
MicroExpert	proc: poor. decl: poor.	proc: poor. decl: poor.	proc: poor. decl: poor.	proc: excellent. decl: excellent.	poor.
Neural Networks, K-Lines, Inference Networks.	neurons, connections, activation, inhibition, thresholds, rules for signal spreading.	concepts - neurons; relationships - connections; inferences - changes in strength.	neural hierarchies, K-Lines.	amplification - activation, inhibition, thresholds; or probabilistic inference.	implementation effort: high. development effort: high. run time efficiency: low. worst feature: semantic representation. best feature: activation based control.

Figure 6 - Comparison of tools and methodologies and their suitability for *OOPS*

Finally, conceptual graphs show the lead in all aspects of knowledge representation, manipulation and selection. Their processing mechanisms are usually organized around pattern matching or are inference-based.

Estimates of the tools' implementation and development efforts and their run time efficiency, is based on the results collected by other researchers (NIWA, SASAKI

AND IHARA 1984). Niwa and others established frame systems, here PEARL, to be efficient, fast and insensitive to expanding knowledge volume; they are, however, very hard to implement and use (especially for unstructured domains of knowledge). Systems requiring mathematical completeness, as is the case with PROLOG, are the slowest, also difficult to implement and very sensitive to the changes in knowledge volume. Actor or production systems are somewhere in the middle between frames and logic systems. We believe that SCHEME and other LISPs are in general the most efficient but expressively poorest tools. On the other end of the scale we have conceptual graphs which have very flexible representation capabilities, but significantly poorer performance than frame systems, due to their conceptual complexity. However, the reducibility of conceptual and procedural expressions to primitives in conceptual graphs suggests a possibility of efficient compiled encoding of the system's manipulation and inferencing procedures. Neural networks are quite an expensive tool to develop; their efficiency could also be expected to be poor (learning and subsequent recall require long iterative relaxation methods).

A trade-off between structure, modularity, expressive power, and efficiency could be negotiated by combining features of conceptually different tools. Frames could be coupled with production rules (as in LOOPS and KEE - FIKES AND KEHLER 1985), frames with logics (as in KRYPTON - BRACHMAN, FIKES AND LEVESQUE 1983), or in conceptual graphs which combine highly formal graph structures with actor formalisms (SOWA 1984). It could be seen that a combination of conceptual graphs and neural networks may result in an excellent tool for OOPS.

At the time of tool evaluation, and also at the time of writing this paper, there was no conceptual graph processor available to this research; also the inference network system MICROEXPERT was found not to be flexible enough to accommodate any significant development and extension. The best options seemed to start from either a frame system (PEARL) or from an actor system (NEOACTORS) and to slowly work towards a system being a combination of conceptual graphs, with the added element of activation spreading. After a closer investigation, PEARL was found to have a number of peculiarities that could be attributed to its efficient storage and manipulation techniques, and which could cause some major drawbacks during later PEARL utilization (e.g. a limitation on the frame size, strange and awkward multi-frame matching using freezing and thawing of unification stack, etc…). NEOACTORS offered a full implementation of message passing between frame-like clusters of a very simple nature. It was decided to adopt certain elements of this system, i.e. its message passing capabilities and process co-routining,[3] and to redevelop its frame structure to represent complex semantic networks. Also an agenda-based co-routining of NEOACTORS was used as a basis for the implementation of neural-like activation spreading; the agenda priorities are calculated as the number of requests pending. Such an approach alleviated the burden of LISP programming from its lowest primitives. NEOACTORS is a FRANZLISP shell and thus the knowledge clustering capabilities could not be based on lambda closures (due to the FRANZLISP dynamic scoping); instead a-lists and p-lists were taken as a clusters' fundamental data structure.

7 Conclusions

This report defined a concept of "object-orientation". It identified the major components of an object-oriented system applicable to represent and manipulate a robot world, namely the mapping (primitives), organization (concepts, relationships and procedures), clustering (prototypical taxonomies), and finally selection (activation mechanism). We reviewed a number of tools representing different knowledge representation models - functional (SCHEME), logical - (PROLOG), frames (PEARL), actors (NEOACTORS), integrated (Conceptual Structure Processor), and activation networks (MICROEXPERT). Tha paper provided criteria to compare and contrast the systems and their models, based on which NEOACTORS was considered to be the only available system capable of further development into a hybrid system combining the powers of conceptual graphs and those of an activation model.

The resulting system, called CONTEXTUS was then developed and still being under investigation, retains NEOACTORS' message passing system and its co-routining based on a process agenda. The actors' memory organization had to be adjusted to cater for storing complex semantic networks and actors' procedures at the same time. The neural-like activation and inhibition was to be implemented via a process agenda and its priorities; the priorities are based on the number of pending requests.

A more detailed architecture and workings of the system developed from this analysis could be referred to CYBULSKI 1987.

---

3 Note that NEOACTORS was also developed as a part of this research.

Bibliography

ABELSON, H., SUSSMAN, G.J. AND SUSSMAN, J. (1985): *Structure and Interpretation of Computer Programs,* Cambridge, MA: The MIT Press.

AGHA, G. (1986): *ACTORS: A Model of Concurrent Computation in Distributed Systems,* Cambridge, MA: The MIT Press.

AGHA, G. AND HEWITT, C. (1987): "ACTORS: A Conceptual Foundation for Concurrent Object-Oriented Programming," in SHRIVER AND WEGNER 1987, 49-74.

ALLEN, E.M., TRIGG, R.H. AND WOOD, R.J. (1985): *The Maryland Artificial Intelligence Group Franz Lisp Environment,* Maryland Artificial Intelligence Group, Baltimore, MD: University of Maryland.

ANDERSON, J.A. AND ROSENFELD, E. (EDS 1988): *Neurocomputing: Foundations of Research,* Cambridge Massachusetts: The MIT Press.

BACH, E. AND HARMS, R.T. (ED 1968): *Universals in Linguistic Theory,* Chicago, Ill: Holt, Rinehart and Winston.

BARTLETT, F.C. (1932): *Remembering: A Study in Experimental and Social Psychology,* Cambridge, England: The University Press.

BOBROW, D.G. AND COLLINS, A. (ED 1975): *Representation and Understanding: Studies in Cognitive Science,* New York, NY: Academic Press.

BOBROW, D.G., KAPLAN, R.M., KAY, M., NORMAN, D.A., THOMPSON, H. AND WINOGRAD, T. (1977): "GUS, A Frame-Driven Dialog System," *Artificial Intelligence **8**,* 155-173.

BOBROW, D.G. AND WINOGRAD, T. (1977): "An Overview of KRL, a Knowledge Representation Language," *Cognitive Science **1**,* 3-46.

BOBROW, D.G., KAHN, K., KICZALES, G., MASINTER, L., STEFIK, M. AND ZDYBEL, F. (1985): *COMMONLOOPS: Merging COOMMON LISP and Object Oriented Programming,* Rep. ISL-85-8, Palo Alto, CA: Xerox Palo Alto Research Center.

BORGIDA, A., MYLOPOULOS, J. AND WONG, H.K.T. (1984): "Generalization/Specialization as a Basis for Software Specification," in BRODIE, MYLOPOULOS, SCHMIDT 1984, 87-114.

BRACHMAN, R.J. (1983): "What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks," *IEEE Computer **10***, 30-36.

BRACHMAN, R.J. (1985): "I Lied About the Trees," *AI Magazine **3**,* 80-93.

BRACHMAN, R.J. AND SCHMOLZE, J.G. (1985): "An Overview of KL-ONE Knowledge Representation System," *Cognitive Science **9**,* 171-216.

BRACHMAN, R.J., FIKES, R.E. AND LEVESQUE, H.J. (1983): "KRYPTON: A Functional Approach to Knowledge Representation," *IEEE Computer **16**,* 67-73.

BRODIE, M.L., MYLOPOULOS, J. AND SCHMIDT, J.W. (ED 1984): *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases and Programming Languages,* New York, NY: Springer-Verlag.

BRUCE, B. (1973): "Case Structure Systems," *IJCAI **73**,* 364-371.

BYTE (1986): *Special Issue on Object-Oriented Languages,* August.

CARBONELL, J.G. (1978): "Politics: Automated Ideological Reasoning," *Cognitive Science **1***, 27-51.

CLOCKSIN, W.F. AND MELLISH, C.S. (1981): *Programming in Prolog,* Berlin, Heidelberg: Springer-Verlag.

COLLINS, A.M. AND LOFTUS, E.F. (1975): "A Spreading Activation Theory of Semantic Processing," *Psychological Review **6***, 407-428.

COX, B.J. (1987): *Object Oriented Programming: An Evolutionary Approach,* Reading Massachusetts, Addison Wesley.

COX, P.R. AND BROUGHTON, R.K. (1982): *Micro Expert Users Manual, Version 2.1.1,* ISIS Systems Ltd.

CULLINGFORD, R.E. (1978): *Script Application: Computer Understanding of newspaper stories,* Ph.D. Dissertation, Research Report 116, Computer Science Department, Yale University.

CYBULSKI, J.L. (1987): *Development of Concepts and Methodologies for the Representation of Contextual Information in Knowledge Based Systems*, Masters Thesis, Department of Computer Science, Melbourne:RMIT.

DAVIS, R. AND KING, J.J. (1977): "An Overview of Production Systems," in ELCOCK AND MICHIE 1977, 300-332.

DAVIS, R. AND LENAT, D.B. (1980): *Knowledge Based Systems in Artificial Intelligence,* New York, NY: McGraw-Hill.

DEERING, M., FALETTI, J. AND WILENSKY, R. (1982): *Using the PEARL AI Package,* Computer Science Division, University of California, Berkeley.

DEJONG, G.F. (1979): *Skimming Stories in Real Time: An Experiment in Integrated Understanding,* Ph.D. Dissertation, Research Report 158, Department of Computer Science, Yale University.

DUDA, R.O., HART, P.E., NILLSON, N.J. AND SUTHERLAND, G.L. (1978): "Semantic Network Representations in Rule-Based Inference Systems," in WATERMAN AND HAYES-ROTH 1978, 203-221.

DYER, M.G. (1983): *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension,* Cambridge, MA: The MIT Press.

ELCOCK, E.W. AND MICHIE, D. (EDS 1977): *Machine Intelligence **8***, Chichester, England: Ellis Horwood.

ERMAN, L.D., HAYES-ROTH, F., LESSER, V. AND REDDY, D. (1980): "The HEARSAY-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty," *Computing Surveys **12***, 409-415.

FAHLMAN, S.E. (1979): *NETL: A System for Representing and Using Real-World Knowledge,* Cambridge, MA: The MIT Press.

FALETTI, J. AND WILENSKY, R. (1982): *The Implementation of PEARL,* Computer Science Division, University of California, Berkeley.

FARGUES, J., LANDAU, M.C., DUGOURD, A. AND CATACH, L. (1986): "Conceptual Graphs for Semantics and Knowledge Processing," *IBM Journal of Research and Development **1***, 70-79.

FELDMAN, J.A. (1986): "Connections," *Byte **4***, 277-285.

FIKES, R.E. AND KEHLER, T. (1985): "The Role of Frame-Based Representation in Reasoning," *Communications of the ACM **9**,* 904-920.

FILLMORE, C. (1968): "The Case for Case," in BACH AND HARMS 1968, 1-90.

FINDLER, N.V. (ED 1978A): *Associative Networks: The Representation and Use of Knowledge in Computers,* Academic Press.

FINDLER, N.V. (1978B): "A Heuristic Information Retrieval System Based on Associative Networks," in FINDLER 1978A, 305-326.

FORGY, C. AND MCDERMOT, J. (1977): "OPS: A Domain-Independent Production System Language," *IJCAI **5**,* 933-939.

GOLDBERG, A. AND ROBSON, D. (1983): *Smalltalk-80: The Language and Its Implementation,* Reading, MA: Addison-Wesley.

GOLDSTEIN, I.P. AND ROBERTS, B. (1979): "Using Frames for Scheduling," in WINSTON AND BROWN 1979, 255-286.

HAYES, P.H. (1977): "On Semantic Nets, Frames and Associations," *Proc. IJCAI **5**,* 99-107.

HAYES, P.J. (1979): "The Logic of Frames," in METZING 1979, 46-61.

HEBB, D.O. (1949): *The Organization of Behaviour,* New York, NY: Wiley.

Also ANDERSON AND ROSENFELD 1988, 45-56.

HENDRIX, G. (1978): "Semantic Knowledge," in WALKER 1978, 121-228.

HEWITT, C. (1971): *Description and Theoretical Analysis (using schemas) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot,* Doctoral Dissertation, AI Lab, Cambrideg, MA: MIT.

HEWITT, C. (1977): "Viewing Control Structures as Patterns of Passing Messages," *Artificial Intelligence **8**,* 323-364.

HEWITT, C., BISHOP, P. AND STEIGER, R. (1973): "A Universal Modular ACTOR Formalism for Artificial Intelligence," *IJCAI **73**,* 235-245.

HEWITT, C.E. AND DE JONG, P. (1984): "Open Systems," in BRODIE, MYLOPOULOS AND SCHMIDT 1984, 147-164.

HEWITT, C.E., REINHARDT, T., AGHA, G. AND ATTARDI, G. (1985): "Linguistic Support of Serializers for Shared Resources," in *Seminar on Concurrency,* 330-359, Springer-Verlag.

HINTON, G.E. (1985): "Learning in Parallel Networks," *Byte **4**,* 265-276.

JORGENSEN, C.C. AND MATHEUS, C. (1986): "Catching Knowledge in Neural Nets," *AI Expert **4**,* 30-41.

KING, M. (ED 1983): *Parsing Natural Language,* Academic Press.

KUHN, T. (1970): *The Structure of Scientific Revolution,* 2nd ed., Chicago, Ill: University of Chicago Press.

KUIPERS, B.J. (1975): "A Frame for Frame: Representing Knowledge for Recognition," in BOBROW AND COLLINS 1975, 151-184.

KUNZ, J., KEHLER, T.P., AND WILLIAMS, M.D. (1984): "Applications Development Using a Hybrid AI Development System," *The AI Magazine* ***3***, 41-54.

LINDSAY, P.H. AND NORMAN, D.A. (1977): *Human Information Processing: An Introduction to Psychology,* New York, NY: Academic Press.

MAMDANI, E.H. AND GAINES, B.R. (1981): *Fuzzy Reasoning and Its Applications,* New York, NY: Academic Press.

MARTIN, W.A. (1979): "Descriptions and the Specialization of Concepts," in WINSTON AND BROWN 1979, 375-420.

MCCLELLAND, J.L., RUMELHART, D.E. AND HINTON, G.E. (1986): "The Appeal of Parallel Distributed Processing," in RUMELHART AND MCCLELLAND 1986, 3-44.

MCCULLOCH, W.S. AND PITTS, W.H. (1943): "A Logical Calculus of Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics* ***5***, 115-133.

Also ANDERSON AND ROSENFELD 1988, 18-28.

MEEHAN, J. (1976): *The Metanovel: Writing Stories by Computer,* Ph.D. Dissertation, Research Report 74, Computer Science Department, Yale University.

METZING, D. (ED 1979): *Frame Conceptions and Text Understanding,* Berlin: Walter de Gruyter and Co.

MICHALSKY, R.S. AND CHILAUSKY, R.L. (1981): "Knowledge Acquisition by Encoding Expert Rules Versus Computer Induction From Examples: A Case Study Involving Soybean Pathology," in MAMDANI AND GAINES 1981, 247-272.

MINSKY, M. (ED 1968): *Semantic Information Processing,* Cambridge, MA: MIT Press.

MINSKI, M. (1975): "A Framework for Representing Knowledge," in WINSTON 1975, 211-280.

MINSKY, M. (1979): "The Society Theory of Thinking," in WINSTON AND BROWN 1979, 422-450.

MINSKY, M. (1981): "K-Lines: A Theory of Memory," in NORMAN 1981, 87-101.

MINSKY, M. (1986): *The Society of Mind,* New York, NY: Simon and Shuster.

MINSKY, M. AND PAPERT, S. (1969): *Perceptrons,* Cambridge, MA: The MIT Press.

MYLOPOULOS, J., BERNSTEIN, P.A. AND WONG, H.K.T. (1980): "A Language Facility for Designing Database-Intensive Applications," *ACM Trans. on Database Systems* ***2***, 185-207.

MYLOPOULOS, J., COHEN, P., BORGIDA, A. AND SUGAR, L. (1976): "Semantic Networks and the Generation of Context," *IJCAI 76,* 134-142.

NII, H.P., FEIGENBAUM, E.A., ANTON, J.J. AND ROCKMORE, A.J. (1982): "Signal to Symbol Transformation: HASP/SIAP Case Study," *AI Magazine* ***3***, 23-35.

NII, H.P. (1986A): "The Blackboard Model of Problem Solving," *AI Magazine* ***2***, 38-53.

NII, H.P. (1986B): "Blackboard Systems Part Two: Backboard Application Systems," *AI Magazine* ***3***, 82-106.

NIWA, K., SASAKI, K. AND IHARA, H. (1984): "An Experimental Comparison of Knowledge Representation Schemes," *AI Magazine* ***2***, 29-36.

NORMAN, D.A. (1969): *Memory and Attention: An Introduction to Human Information Processing,* New York, NY: Wiley & Sons.

NORMAN, D.A. (ED 1981): *Perspectives on Cognitive Science,* Lawrence Erlbaum.

NORMAN, D.A. AND BOBROW, D.G. (1975): "Some Principles of Memory Schemata," in BOBROW AND COLLINS 1975, 131-150.

PETERSON, J.L. (1981): *Petri Net Theory and the Modelling of Systems,* Englewood Cliffs, NJ: Prentice-Hall.

POLLACK, J.B. AND WALTZ, D.L. (1984): "Parallel Interpretation of Natural Language," *Proc. of The International Conference on Fifth Generation Computer Systems,* ICOT.

PUTNAM, H. (1962A): "The Analytic and the Synthetic," in PUTNAM 1962, 33-69.

PUTNAM, H. (1962B): *Mind, Language, and Reality,* Cambridge: Cambridge University Press.

QUILLIAN, M.R. (1968): "Semantic Memory," in MINSKY 1968, 227-270.

REITER, J. (1981): *AL/X: An Inference System for Probabilistic Reasoning,* Msc Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign.

ROBERTS, R.B. AND GOLDSTEIN, I.P. (1977): *The FRL Primer,* Rep. AIM-408, AI Lab, Cambridge, MA: MIT.

ROSENBLATT, F. (1958): "The Perceptron: A Probabilistic Model for Information Storage and Retrieval in the Brain," *Psychological Review **65***, 386-408.

Also ANDERSON AND ROSENFELD 1988, 92-114.

RUMELHART, D.E. (1975): "Notes on a Schema for Stories," in BOBROW AND COLLINS 1975, 211-236.

RUMELHART, D.E. AND NORMAN, D.A. (1973): "Active Semantic Networks as a model of Human Memory," *IJCAI **73**,* 450-457.

RUMELHART, D.E. AND MCCLELLAND, J.L. (ED 1986): *Parallel Distributed Processing Explorations in the Microstructure of Cognition,* Cambridge, MA: The MIT Press.

RUMELHART, D.E., HINTON, G.E. AND MCCLELLAND, J.L. (1986): "A General Framework for Parallel Distributed Processing," in RUMELHART AND MCCLELLAND 1986, 45-76.

SCHANK, R.C. (1975): *Conceptual Information Processing,* Amsterdam, Holland: North-Holland.

SCHANK, R.C. AND ABELSON, R.P. (1977): *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures,* Hillsdale, NJ: Lawrence Erlbaum Associates.

SCHANK, R.C. AND COLBY, K.M. (1973): *Computer Models of Thought and Language,* San Francisco, CA: Freeman.

SCHANK, R.C., LEBOWITZ, M. AND BIRNBAUM, L. (1978): *Integrated Partial Parsing,* Research Report 143, Computer Science Department, Yale University.

SCHANK, R.C. AND RIESBECK, C.K. (1981): *Inside Computer Understanding: Five Programs Plus Miniatures,* Hillsdale, NJ: Lawrence Erlbaum.

SHORTLIFFE, E.H. (1976): *Computer Based Medical Consultations: MYCIN,* New York, NY: Elsevier North Holland.

SHRIVER, B. AND WEGNER, P. (EDS 1987): *Research Directions in Object-Oriented Programming,* Cambridge, Massachusetts: The MIT Press.

SIMMONS, R.F. (1973): "Semantic Networks: Their Computation and Use for Understanding English Sentences," in SCHANK AND COLBY 1973, 63-113.

SMALL, S., COTTRELL, G. AND SHASTRI, L. (1982): "Toward Connectionist Parsing," *Proc. AAAI **82**,* 247-250.

SOWA, J.F. (1984): *Conceptual Structures*, Reading, MA: Addison-Wesley Pub. Co.

STEFIK, M. (1979): "An Examination of a Frame-Structured Representation System," *IJCAI **79**,* 845-852.

STEFIK, M. (1980): *Planning with Constraints,* Rep. No. 784, Computer Science Dept., Stanford University.

STEFIK, M. AND BOBROW, D. (1986): "Object-Oriented Programming: Themes and Variations," *The AI Magazine **4**,* 40-62.

STEFIK, M., BOBROW, D., MITTAL, S. AND CONWAY, L. (1983): "Knowledge Programming in LOOPS," *The AI Magazine **3***, 3-14.

TERRY, A. (1983): *The CHRYSALIS Project: Hierarchical Control of Production Systems,* Tech. Rep. HPP-83-19, Stanford University, Heuristic Programming Project.

THERIAULT, D. (1982): *A Primer for the ACT-1 Language,* A.I. Memo 672, AI Lab, Cambridge, MA: MIT.

THERIAULT, D. (1983): *Issues in the Design and Implementation of Act2,* Technical Report 728, AI Lab, Cambridge, MA: MIT.

WALKER, D.E. (ED 1978): *Understanding Spoken Language,* New York, NY: North-Holland.

WASSERMAN, K. (1985): "Physical Object Representation and Generalization: A Survey of Programs for Semantic-Based Natural Language Processing," *The AI Magazine **4**,* 28-43.

WATERMAN, D.A. AND HAYES-ROTH, F. (EDS 1978): *Pattern-Directed Inference Systems,* New York, NY: Academic Press.

WILENSKY, R. (1978): *Understanding Goal-Based Stories,* Ph.D. Dissertation, Research Report 140, Computer Science Department, Yale University.

WILENSKY, R. (1984): *LISPcraft,* New York, NY: Norton.

WILKS, Y. (1973): "Understanding Without Proofs," *IJCAI **73**,* 270-277.

WINOGRAD, T. (1975): "Frame Representations and the Declarative-Procedural Controversy," in BOBROW AND COLLINS 1975, 185-210.

WINSTON, P.H. (ED 1975): *The Psychology of Computer Vision,* New York, NY: McGraw-Hill.

WINSTON, P.H. (1979): "Learning by Creating Transfer Frames," in WINSTON AND BROWN 1979, 347-376.

WINSTON, P.H. (1984): *Artificial Intelligence, Second Edition,* Readings, MA: Addison-Wesley.

WINSTON, P.H. AND BROWN, R.H. (1979): *Artificial Intelligence: An MIT Perspective,* Cambridge, MA: The MIT Press.

WOODS, W.A. (1975): "What's in a Link: Foundations for Semantic Networks," in BOBROW AND COLLINS 1975, 35-82.