A PANEL ON KNOWLEDGE REPRESENTATION

chaired by
Daniel G. Bobrow
Xerox Palo Alto Research Center
Palo Alto, California

A basic premise of current work in Artificial Intelligence is that the intelligence of systems is a function of the knowledge they contain, and that the utility of this knowledge is in part dependent on the form of its representation. In order to compare different forms of knowledge representation, I asked each of the panel members listed below to respond briefly to three questions. In the little space available, there was obviously no way of exploring these questions in any detail, or to explore more than a personal view of a particular approach. References are provided to more extended discussions in relevant papers. The reader should expect to use this paper as a pointer to what each panel member thinks is important, *not* as an introduction to the material discussed.

Participants and hobby horses

Daniel G. Bobrow — Xerox Palo Research Center
KRL, A Knowledge Representation Language

Gary G. Hendrix — SRI
Partitioned Semantic Nets

William A. Martin — **MIT**
OWL

John McCarthy — **Stanford University**
Predicate Calculus

Allen Newell — **Carnegie Mellon University**
Production Systems

Roger Schank — **Yale University**
Scripts, Plans, and Knowledge Structures

Brian C. Smith — **MIT**
Knowledge Representation Semantics

N. S. Sridharan — **Rutgers University**
AIMDS

The three questions

What are the most important premises underlying your approach to knowledge representation, the critical ideas, and major mechanisms used in your system?

If your representation were being used as a basis for a system which would conduct typed English dialogs with a user about some subject, what aspects would your knowcldge representation make easiest; what aspects would best be handled by building additional mechanisms?

What problem illustrates what you believe your system is best at, and is difficult for some representations? Point out which of your premises and/or ideas make it possible to handle your problem cleanly.

Some commentary on the responses

In my view, the purpose of a panel is to allow those attending to get a sense of the issues of concern and the style of attack of the different participants. A coherent presentation is absent because of the diversity of views and values about what is important. This paper has the same flavor. The following are a number of contrasts that 1 became aware of while reading the individual sections.

1. Some systems place high value on being able to get into and out of English easily (OWL), and others ignore that issue (Production Systems)
2. Some systems emphasize uniformity of representational structure, with user encoding within that form (Predicate Calculus), while others provide different specialized forms for different kinds of knowledge (Schank's Knowledge Structures)
3. Some approaches stress the desire for a clear underlying semantics of representation (KRS) versus others which emphasize empirical adequacy of representation in experiments (Production Systems).
4. Some systems stress logical consistency and formal inference (Predicate Calculus) while others emphasize "reasonable" inference, and heuristic adequacy (KRL)
5. Some systems stress control of access to information (KRL) while others which just ignore that issue (Predicate Calculus)
6. Some systems stress multiple levels of scoping (Partitioned Semantic Nets) while others provide no scoping at all (Production Systems).
7. Some systems allow attachment of special purpose procedural information to the declarative structures (KRL), while some maintain a complete separation of declarative and procedural aspects (Predicate Calculus).
8. Some systems stress dealing with changing world models (AIMDS) while others take a purely local view of changes (Production Systems).

I have have made each of these points as a binary distinction, and the exemplars I chose are those I believe represent extreme positions. There is a spectrum in each case, and different practitioners of the same art of representation move along that spectrum. For example, Kowalski has a system for doing Predicate Calculus inferences which deals quite explicitly with access issues. Finally, not all of the participants would agree with my evaluations of the place they or their systems are in this spectrum, and one must look to these points as guidance for issues to be considered, not an ultimate characterization of systems of the type described.

KRL, A KNOWLEDGE REPRESENTATION LANGUAGE

Daniel G. Bobrow*

* The ideas described here are the result of collaboration with Terry Winograd; we have profiled from interactions with many other people, including Martin Kay, Ron Kaplan. David Levy, Mitch Model, Don Norman, Brian Smith, and Henry Thompson. None of them is to be blamed for my expression of the ideas.

KRL is intended to be a knowledge representation language with which to build sophisticated systems and theories of language understanding. A first implementation, KRL-0 (Bobrow and Winograd, 1977) was used for a number of test programs, and its problems (Bobrow, Winograd and the KRL Research Group, 1977) led to a redesign now being implemented as KRL-l. The major premises of KRL are:

1. Knowledge should be organized around conceptual entities with associated descriptions and procedures.

2. A description must be able to represent partial knowledge about an entity and accommodate multiple descriptors which can describe the associated entity from different viewpoints.

3. An important method of description is comparison with a known entity, with further specification of the described instance with respect to the prototype.

4. Descriptions are legitimate entities in the system, and thus can be described. A system ought to be able to describe all its own operations and structures.

5. Reasoning is dominated by a process of recognition in which new objects and events are compared to stored sets of expected prototypes, and in which specialized reasoning strategies are keyed to these prototypes.

6. Intelligent programs will require multiple active processes with explicit user-provided scheduling and resource allocation heuristics.

7. Information should be clustered to reflect use in processes whose results are affected by resource limitation and differences in information accessibility.

8. A knowledge representation language must provide a flexible set of underlying tools, rather than embody specific commitments about either processing strategies or the representation of specific areas of knowledge.

9. A knowledge representation system ought to have a clear well-defined semantics, where "semantics" is used here as parallel to the sense of semantics used by Tarski for logic. KRS (see the section by Brian Smith) is intended to provide a clean semantic basis for KRL.

The basic mechanisms in KRL include:

1. Scoped units of memory, with costs associated with traversing scope boundaries; these interact with a family of procedures to add information within a scope, and to seek and match beliefs found within and across scopes.

2. Methods for associating procedural information with entities, and invocation of procedures dependent on specific actions on those entities. Procedures are controlled through a multiprocess agenda which can be scheduled by the user. Schedulers use information about both current goals and resource availability.

3. An active belief context which acts as an attention focussing mechanism. It is this active context on which the inference mechanism works to derive new beliefs. Incorporating information from memory structures into the active context is a separately controllable process from performing inferences on that set of active beliefs.

4. An indexing mechanism which allows the user to specify how some data items can be addressed by contents, and which parts of a given structure are to act as an access key. Index keys are general descriptions, not just atoms as in simple semantic nets.

5. A means of attaching descriptions to descriptions so that knowledge about knowledge is expressible easily and directly in the same formalism. For example, these "meta-descriptions" can specify when specific information might be useful, or the dependency of a particular description on other assumptions made in reasoning.

6. A mechanism for compiling objects and operations so that the overhead of general forms and interpreter mechanisms can be avoided in simple cases. This depends on a data compaction scheme which allows specialized data structures to be interpreted as full fledged descriptions.

English Dialog

A basic premise of our approach to natural language understanding is that a dialog system demands integration of a number of complex components through many layers of abstraction As a basis one needs a computer representation, programming, monitoring and debugging system. On top of that must be constructed basic reasoning strategies. Specialized representations and reasoning must be provided for common domains such as time, causality, planning, events and states, etc. One also needs linguistic knowledgee.g.,syntax and parsing strategies, mophological rules, discourse structures. Finally, a dialog system must have expert knowledge of the task domain itself, such as travel planning, medical diagnosis etc. Although we intend to implement modules at all these layers, KRL-l will only embody the first two. The designer of a dialog system will have to provide specialized knowledge (built in KRL) for the rest of the layers.

Strengths and Weaknesses

A major strength of KRL is that it is a self-descriptive system. Because of the access compiler and data compaction mechanisms, this self description can be used to actually implement the system itself. This implies that the basis of the system is accessible to the user in a way that has not been true of other knowledge representations since LISP.

A problem for KRL is that our current representation of processes is weak. Giving a sequence of instructions which an interpreter should follow in order is only one way of describing a procedure. We believe that it will be possible to develop a notion of factored description in which a procedure is described through multiple perspectives which may combine high level statements about the structure of the process, its results, conditions on various parts, ways it fits into goal structures, etc. We would like to apply the self descriptive power of the language to use the reasoning, matching and problem solving powers of KRL as fundamental elements in our tools for designing, building, and working with KRL programs.

SOME GENERAL COMMENTS ON SEMANTIC NETWORKS

Gary G. Hendrix

Semantic networks come in many varieties, reflecting numerous independent developments. Broadly speaking, any representation interlinking nodes with arcs could be called a semantic network, including the structures of Quillian (1968), Simmons (1973), Norman and Rumelhart (1975), and Hendrix (1975,1976).

Nets differ in their expressive power and in the types of procedures designed to interpret them, but all share the feature that the structure used to encode information serves also as a guide for information retrieval. From a given node, nodes representing closely related objects (physical objects, events, situations, etc.) are found simply by following arcs (either incoming or outgoing) from the node to its neighbors. In this way, a network provides its own semantic cross-reference system. To the extent that the labels on arcs and nodes are meaningful to net-manipulating procedures, they provide semantic guidance to help traverse the net in search of information relevant to a task.

The problem of generating English expressions for internal symbols illustrates this important feature. Given the information "G25 is a man." and "Boston is the hometown of G25," G25 may be expressed as "a Boston man." But how can the relevant information be found? Using a network, the search from node G25 is straightforward. Without the network's cross-index, much computation may be required.

The indexing ability of networks is enhanced by introducing partitions (Hcndrix, 1975), which allow parts of the net to be distinguished as if shown in different colors. To solve particular types of problems, only structures of an appropriate "color" are considered.

A network, like other representations (including procedural code), is a passive structure which must be manipulated by an external agent. Thus, a net-based computational system must include both a net and a net manipulator. Unlike some systems that force a position on the procedural/declarative tradeoff, networks offer a spectrum of possibilities concerning how much knowledge is to be encoded in the net as opposed to the manipulator.

For example, consider the statement
SI: Vx[P(x) => Q(x)]
This information might be encoded procedurally in the manipulator so that if, for some K, P(K) arises in the net, then Q(K) may be deduced. Rieger (1974), for example, encoded knowledge procedurally about how to draw inferences from the primitives of conceptual dependency. Other workers have developed systems that seek to place maximum knowledge in the net, minimizing knowledge needed in the manipulator. One method for doing this, investigated by Kay (1973), Shapiro (1971), Hendrix (1975,1976), and Schubert (1972), is to develop network formulations having the full expressive power of the predicate calculus (in addition to the practical advantages of indexing). The logically complete systems can encode statement SI directly in the network, allowing them to reason about the statement itself. They may express it in English, or indicate who believes it, or use it in deductions. (Procedural encodings are often only able to show Q(K) given P(K).) The manipulator for a logically complete net may be a relatively simple deductive mechanism based on conventional theorem proving, but algorithms for manipulating nets may move beyond conventional theorem proving and use knowledge about procedures. defaults, fuzzy inference and resource-bound computation.

Given that the manipulator does not manipulate itself, moving knowledge from the manipulator to the net makes it available for inspection and is a step toward generalization and introspection. However, such "deproceduralization" generally reduces efficiency. A solution (which may introduce consistency problems) is to encode some knowledge both ways. Declarative structures in the net may model procedures (including procedures in the manipulator), their hierarchical decomposition into substeps. the effects they impart when executed, and the relationships among these. The procedures (perhaps compiled from net representation) may be used to answer ground-level questions or to link to motor activities. Descriptions of procedures (or of the knowledge they impart) may be used to answer meta questions. They may be manipulated to effectively execute the procedure, to express the sequence of procedural steps in English, to answer questions about the structure of the procedure, and to do planning.

In building a network system to understand English discourse, all the knowledge about language and translation is typically placed in the language portion of the manipulator. The net (and the question answering portion of the manipulator) provide a model of the domain of discourse, which typically excludes language. But interesting new abilities arise if linguistic information moves to the net. The resolution of definite references is made easier if the discourse context (a linguistic notion) is encoded in the network (Grosz. 1977). The processing of elliptical or quantified utterances is made easier by using the network to encode the relationships between syntactic units and the meanings they convey (Grosz 1977, Hendrix 1976). If the lexicon is represented in the net, the system becomes capable of discussing the relationship between concepts and the words that express them. If the parser is represented in the net, the system may answer questions about its structure and about how it parses inputs.

Although semantic networks are a relatively old representation medium, they have been flexible in adapting to new innovations and hold considerable potential for future development

OWL

William A. Martin

I assume the panel members share my belief that development of a comprehensive methodology for the representation of knowledge which can be embodied in a practical programming language is a key direction for AI research at this time. OWL. is distinguished from the other efforts in that it is based explicitly and in detail on the syntactic and semantic structure of English. In launching the owt project I determined 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, OWL is an attempt to explore whether language can also serve as a source of conventions for the development of a computer language. One advantage of this approach is that since owi. is so much like English, the English to OWL and OWL to English translations can be made more easily than is the case for the other systems. At the same time OWL contains many of the same mechanisms of the other systems, so it may achieve this advantage without a corresponding loss, "OWL" stands for "One World Language", reflecting the fact that the conventions of English provide a unifying framework within which each of us develops his thoughts. If this same unifying framework can indeed by transferred to a machine it can perhapsform a basis for achieving wider applicability of independently developed procedures and for absorbing into a computer the knowledge of individual English speaking experts. More immediately, since owt is based on English, our development team is accustomed to settling disputes over what may be equally good representation alternatives by appeal to English.

The most novel aspects of OWL are the structure of the semantic net (Hawkinson. 1975) and the approach to "reference". Nodes of the OWL semantic net are called "concepts". Concepts are written as parenthesized expressions. That is, each node of the OWL semantic net is an expression! In translating English to OWL we establish a mapping between English sentences, phrases, word senses, and affixes and expressions. Eor example, the OWL concept corresponding to "the bucket" is (BUCKET'X THE). Concepts are stored uniquely, like LISP atoms, in the owt semantic net. Each concept has a "reference list", which is like the property list of a LISP atom except that it contains singleton concepts instead of attribute/value pairs.

The expression for any concept, C, is constructed from two other concepts termed the "genus" and "specializer" of C. The general form of the expression is (genus*meta-attribute-abbrcviation specializer). There are seven possible meta-attributes as shown in Figure 1.

Meta-altrlbute abbreviation		example of use
SPECIES	S	bull dog -> (DOG*S BULL)
STEREOTYPE	T	lap dog -> (DOG*T LAP)
INSTANCE	I	Fido -> (DOG*I FIDO)
ASPECT	A	Bob's dog -> (DOG*A BOB
RESTRICTION	R	fat dog -> (DOG*R FAT)
INFLECTION	X	dog -> (DOG*X S)
PARTITIVE	P	pack of dogs -> (PACK*P (DOG*X S))

Figure I: Meta-attributes in OWL

By convention, every concept inherits the information on the reference list of its genus, whenever that is not contradicted by information on the reference list of the concept itself. Because of this convention it makes sense to think of concepts as being organized in a hierarchy "under" their genus. The primary role of the special i/er of a concepts. C. is to distinguish C from all other concepts having the same genus and meta-attribute. For example. BULL distinguishes (DOG*S BULL) from all other species of DOG.

Figure 2 shows an example of KRL (Bobrow and Winograd, 1977). The corresponding OWL IS in Figure 3.

[Travel UNIT Abstract
...Travel is the unit name. Its category type is Abstract
<SELF (an EVENT) > ...description of the Travel unit itself.
...Event. Plane etc. are known units
<mode (OR Plane Auto Bus)) ...cither Plane or Auto or Bus
... can fill the slot named mode,
<destination (a City)>]

[Visit UNIT Sperialization ... a specific category of Social Interact ion
<SELF (a SocialIntcraction)>
<visitor (a Person))
<visitecs (SetOf (a Person))>]

(Fventl.V7 UNIT Individual ... a specific event described from two viewpoints
<SELF (a Visit with
visitor - Rusty ...The actor is the known unit Rusty
visilees = (Items Danny Terry)) ... Items indicates at least Danny
(a Travel with ... and Terry are set elements
mode - Plane
destination= SanFrancisco)} >] ..SanFranciso is a known unit

Figure 2. KRL-0 Representation of a Trip to San Francisco

...definition of the verb travel

[TRAVEL = (TRANS.*S TRAVEL")

[SUBJECT.: PERSON]

[MEANS.: (OR PLANE, AUTO, BUS)]

[DESTINATION.: CITY]]

...definition of the noun trip in terms of travel

[TRIP - (TRA\EL*X EVENT-PERFFCTIVF-DFVFRBAL-NOUN.)]

...Definition of the verb visit

[VISIT = (INTERACT-SOCIALLVS -VISIT)

[SUBJECT.: PERSON]

[OBJECT.: PERSON]]

....definition of the noun visit in terms of the verb

[VISIT-EVENT = (VISITX EVENT-PERFFCTIVEDEVERBAL-NOUN.)]

...definition and description of Rusty's trip

[EVENTI37 =
((((VISITT (AND DANNY, TERRY))*T RUSTY)*X

EVENT-PERFECTIVE-DEVERBAL-NOUN.)*! (NO.*S 137))
((TRAVELS (MEANS-BYT PLANE))*T

(DESTINATION-TOT SAN-FRANCISCO))»X

EVENT-PERFEXTIVFDEVERBAL-NOUN.)]

Figure 3. OWL Representation of A Trip to San Francisco

To compare these, one needs to know that in OWL "[A B C]" puts the concepts B and C on the reference list of concept A and has value A. "L = A" gives concept A the label L, which can then be used in referring to it. Also ":" is a macro notation with [CI [C2: C3]] expanding to [CI [(C2*A CI) C3]]. While the owl notation is somewhat more cumbersome than the KRL, it has the advantage that it can be translated to English for debugging, as was done in (Swartout 1977).

It might also be interesting to compare OWL with Schank's conceptual dependency networks. Figure 4 shows two sentences followed by the networks from (Schank 1973) and the corresnondim? OWL.

a) The man took a book.

P o R — to -> man

Schank: man <-> TRANS < book <—|

-- from <- someone

OWL: (MAN*X THF)/(ACTT (PURPOSF-TO.T (CAUSET

((TRANS.T (BOOK*X A))T (DFSTINATION-TO.T -SELF)))))

b) I gave the man a book.

P o R -- to ->man

Schank: I <-> TRANS <book <-|

— from <- I

OWL: I / (ACTT (PURPOSE-TOT (SOURCE.)*T

((TRANS.T(BOOK*X A))T
<DEST|NATION-TO.T(MAN*X THE))))))

Figure 4. Comparison of conceptual dependency and OWL

Because OWL exploits relationships of the type explored by the generative semantics branch of linguistics, both "take" and "give" are expressed in terms of TRANS in OWL, just as in conceptual dependency. However, in OWL, the differing point of view of the two expressions is retained at least in the OWL expression output by the parser. In fact, OWL sides with Chomsky in determining the logical form of a sentence from its surface structure. We have, however, elected not to use the predicate calculus as a way of expressing ambiguities generally associated with quantifier scope. Instead, we take a highly algorithmic view of the process of referent finding.
Consider the expression, "old friend." This is ambiguous: it can mean either a friend who is old or one with whom we have an old friendship. To get the first meaning, "friend" is used to locate the concept FRIEND, the description of FRIEND is then used to locale possible referents of it; these in turn are filtered against the predicate OLD. To get the second reading, "old" and "friend" are used to locate the concept of an old friendship, the description of this concept is then used to locate a referent. In OWL. these two senses are expressed as (FRIEND*R OLD) and (FRIEND*T OLD) respectively. The ambiguous sentence

Betsy wants Sam to read every book that Sally wants him to read.

which has been resolved by quantifier scoping can also be resolved like "old friend". To resolve

Every boy loves a girl on this block.

we distinguish between (GFRL*X (A*S GENERIC)) and (GIRL*X (A*S PARTICULAR)), the generic and particular readings of "a girl" and again avoid explicit quantifier scoping.

PREDICATE CALCULUS

John McCarthy

Q. What are the most important premises underlying your approach to knowledge representation, the critical ideas, and major mechanisms used in your system?

A. At present I am trying to identify the facts about the world that must be used in solving various kinds of problems and the modes of reasoning available to find and validate proposed solutions. An important premise is that the epistemological problem of what knowledge is available to a problem solver with given opportunities to observe and compute is substantially separable from the heuristic problem of how then to decide what to do. In the present stage of research there is no "system" in the sense of a program, although I use our proof-checker to see what the reasoning looks like. It is also a premise, so far verified by experience, that first order logic, i.e. extended predicate calculus, is convenient for expressing these facts about the world. It should be emphasized that first order logic itself does not correspond to a language, it is rather a basic notation within which languages can be developed. Thus if one first order language is found inadequate for some purpose, others with entirely different characteristics can be tried.

Q. If your representation were being used as a basis for a system which would conduct typed English dialogs with a user about some subject, what aspects would your knowledge representation make easiest; what aspects would best be handled by building additional mechanisms.

A. Predicate calculus representations of the knowledge expressed in English is feasible, but more difficult than has been realized in the past. The problem has nothing to do with the syntax of natural languages but with the dependence of the semantics on context. The best work so far in this direction is Richard Montague's "English as a Formal Language" and possibly some of the work of his followers. However, this work does not so far take into account much of what has been accomplished in AI, and I would do many things differently.

While I do not expect to develop a running "Advice Taker" in the near future, I have thought a lot about it. While first order logic formulas in LISP notation would be used to represent some information, most information would be compiled into more purpose-oriented internal forms before use. First order logic might well be used for communicating information.

None of this has much relation to Kowalski's proposals (Kowalski, 1974) to use predicate calculus as a programming language. I agree that this can be done, but I have not yet seen anything to convince me that it has many advantages as a programming language.

Q. What problem illustrates what you believe your system is best at, and is difficult for some representations? Point out which of your premises and/or ideas make it possible to handle your problem cleanly.

A. Since I don't have a "system", it is difficult to respond precisely to the question, bul here are some things that 1 know how to do that I think will offer difficulty to the "systems" that 1 know about.

1. "Travel agents know what flights are available between two cities but don't know the gates from which the airplanes leave. They have general information about the air travel system". A system should be able to receive this fact, in some notation, on its input and know how to find out how to get somewhere, and know enough to ask the travel agent how to find the gate.

2. When asked whether President Carter is standing or sitting at this moment, a program should say it doesn't know, and when asked to think harder, it should say that more thinking wouldn't help, because, as far as its information goes, he could be doing either. This reply should not be made if the program has direct information about his posture.

3. When told that Mary has the same telephone as Mike, that Pat knows Mike's telephone number, and that Pat dialed Mike's telephone number, the program should assent to the statement that Pat dialed Mary's telephone number, and express ignorance about whether Pat knows Mary's telephone number.

4. When told the missionaries and cannibals problem in English or in first order logic, the program should behave differently hearing it as a puzzle and when hearing it when it believes it is silting by the river in a jungle. In the former case it should reject the possibility of a bridge across the river or a lack of oars for the boat. In the latter case, it should find a solution tentatively rejecting them, bul should admit them as possibilities. It needs entities like "a lack of oars" in order to answer questions like "What's wrong with the boal?" and "Is that all that is wrong with the boat?".

My general approach is described in McCarthy and Hayes "Some Philosophical Problems from the Standpoint of Artificial Intelligence", but most of what I have just said is based on three as yet unpublished memoranda "First Order Theories of Individual Concepts", "Minimal Reasoning - A Way of Jumping to Conclusions", and "Ascribing Mental Qualities to Machines".

KNOWLEDGE REPRESENTATION ASPECTS OF PRODUCTION SYSTEMS*

Allen Newell

• This research was supported by the Defense Advanced Research Projects Agency under Contract No. F44620-73-C-0074 and monitored by the Air Force Office of Scientific Research. Charles Forgy. Pat Langley. John McDermott, Kamesh Ramakrishna, and Mike Rychener share credit for the ideas here, but are not to blame for my expression of them.

Production system architectures (PSAs) exhibit too much diversity, both of structure and of purpose, to be considered as a whole. One class of architectures are described here, with roots in human problem solving (Newell & Simon, 72) and cognitive architecture (Klahr & Wallace, 76; Newell, 73), but under exploration as a general architecture for intelligent systems (Rychener, 76; Rychener & Newell, 77). A current realization is a PS language called OPS (Forgy & McDermott, 77), an outgrowth of several earlier CMU PSAs.

Basic architecture (except for input/output)

1. Working memory holds a set of data elements, which arc list structures on atoms. It is a temporary memory, used for focussing attention, holding context, and communicating operands and results.

2. Productions are condition-action pairs. The conditions are forms on working memory elements; the actions are unconditional sequences of additions to working memory, list operations on working memory elements, and constructions of new productions.

3. Production memory hold the set of productions. It is the only permanent memory, i.e., all long term knowledge is stored as productions.

4. The recognition-act cycle repeaiedly evokes a single satisfied production, thus producing a serial stream of actions. The cycle is taken as a basic cycle in a machine, so that many of them may be expected to occur in performing a task.

5. Conflict resolution is the selection within each recognize-act cycle of the production to evoke from the set of satisfied productions. It is not a locus of intelligent selection, but realizes basic system features, such as attention focussing, interruptibihty, instability control (McDermott & Forgy, 77).

6. Recognition match. Under suitable match algorithms and conflict resolution rules, the cycle time is essentially independent of the sizes of production memory and working memory (although bilinear under the naive algorithm). The cycle time does depend on the actions taken by the evoked production (Forgy, 77; McDermott, Newell & Moore, 76).

Such a match can be called a "recognition match", since externally the action appears to be "immediately recognized" rather than the result of extended computation. It is a necessary characteristic of the architecture.

How is knowledge encoded?

7. No syntactic or structural rules, other than list structuring, govern the encoding of knowledge in data elements. (A systematic bias toward prefixing does come from the OPS match, since it allows segment variables only for tails of data elements).

8. Homogeneous encoding: All knowledge in the PS is encoded in the same way, as productions.

9. Grain size: The production is the largest structural unit of knowledge recognized by the architecture. It may be taken as a conjunction of assertions (each action of the action side producing a new data element that may be used independently by other productions). Larger units of data are to be composed of sets of productions, linked together implicitly through their conditions.

IO.PSs of any complexity appear to require a goal structure. Goals are data elements and form a goal-subgoal means-ends network (by linkages through conditions, as per 9). Further experience might reveal alternatives to this organization, but probably not.

What are the key features of the read (access) processes?

II.The access mechanism of the PSA representation is the recognition itself. Its power is determined by the power of a recognition match. There is no adequate characterization of this power. The current logic admits: variables, negated elements, occurrence conditions, multiple conditions applied to the same element, and a single segment variable. Indirect accessing occurs through a sequence of cycles.

12.Wide-band access: The totality of the knowledge is potentially available at any cycle (i.e., all satisfied productions are candidates at each cycle). (Note however that not all knowledge will be accessed when appropriate, since the access relationship is limited, per 11.)

13. No structural separations are enforced (eg, for contexts, for separate worlds, for protection). (This is a consequence of 12, but is worth stating separately.)

What are the key properties of the write (augmentation) processes?

14. Simple addition. Knowledge augmentation is achieved by creating productions and simply adding them to the set of productions in production memory.

An important advance has been to attain conflict resolution schemes that admit this simple rule and do not require positioning the new production vis a vis existing productions, thus requring knowledge of the existing set (McDermott & Forgy, 77).

15. The addition of any knowledge requires the specification of an access path, since the production is a condition-action pair.

The power of the match governs the access relations that can be specified. Thus the simple addition referred to in 14 is not as simple as possible (eg, simply a bin of knowledge items without any indexing); the simplicity arises from having to attend only to the local context of the knowledge to be added and its expected use, not to its relationships with other knowledge.

What about strengths and limitations?

16. PSAs are an interesting candidate for the architecture for human cognition. This is a strength for an architecture for general intelligence only if the (current) architecture can be refined to incorporate human solutions to obtaining general intelligence.

17. PSAs are adequate (and comparable to other AI langages) for programming the range of tasks that AI reasoning programs have investigated, from heuristic search to the blocks world to simple natural language (Rychener, 76). (This statement is appropriate here because doubts have been expressed on this score, PSAs seeming to be useful only for low level processing.)

18. Viewed simply as an architecture to do a circumscribed AI task, there seems no reason to use a PSA unless (1) extremely high conditionally exists (which is what the wide-band access is responsive to) and (2) this access is too irregular to be realized by table-lookup schemes. (This assumes PSAs realized within current machine architectures where the recognition-act cycle imposes some cost.)

19.The greatest limitations lie not in what other representations already do well, but in attaining functions that seem to be latent in the current architecture, e.g., the use of wide-band access to obtain serendipitous access of diverse knowledge, or the full exploitation of simple addition of productions by appropriate debugging and post-adaptation.

CONCEPTUAL DEPENDENCY, AND KNOWLEDGE STRUCTURES

Roger Schank

These are the important premises underlying the knowledge representation theory comprised of Conceptual Dependency, Scripts, Plans, Goals and Themes and other knowledge structures (hence CD/KS).

1. A knowledge representation (KR) should be language-free: that is, it should reflect the important properties of relationships inherent in the world rather than those inherent in the constructor of the KR's native language.

the same event). One of the consequences of this premise is primitive actions, states, objects, etc.

3. A KR should facilitate inference of implicit facts and events. CD/KS provides a canonical form for the representation of events and the intentions that motivate those events. Stored with prototypical canonical forms are rules about reasonable inferences that can be made when those forms are recognized. In this way the representational structures in CD/KS control inference generation.

4. A KR must provide standard event sequences (scripts and named plans in CD/KS) to match input events against so as to determine implicit events. The unique canonical representation used in CD/KS allows us to provide these standard event sequences with easily specificable recognition processes. If we had alternative representational possibilities in the same system we would have difficulty recognizing instances of inputs that triggered or matched our standard event sequences.

5. A KR should provide different depths of detail in representation in a standardized way. In CD/KS there are those clearly specificable levels of representation: the intentional or planning level, the macro-event level for describing actions superficially, and the micro-CD level of describing the details that make up events. Standard event sequences can work at each of these levels, but the distinction of what input triggers what level must be carefully maintained.

6. A KR should facilitiate the process of mapping from a natural language into it and back out of it into a natural language. No information should be lost in this process, and the process itself should exploit the properties of the KR. Any parsing system that fails to exploit the properties of the KR is losing available valuable information.

7. A KR must be predictive. It must have available to it knowledge of standard packages of goals and their realization such that it can predict future inputs on the basis of past inputs. It must also have the ability to switch to bottom-up mode when predictions fail.

II.

At the moment our English input and output system deals with newspaper stories. It handles all facets of the problem when the stories are relatively simple; that is we do question-answering, summary, paraphrase and translation. We are currently building programs to handle more complex domains, but it is clear that the problem at the moment is simply understanding what knowledge is being utilized in complex stories about complex domains. Similarly in a conversational system, the problem is knowing what knowledge people have about the rules of conversation.

Our representation facilitates:

(1) Establishing causal connections

Because there are only a small number of types of connections and the total combination of primitive actions connected according to those types is still rather small;

(2) Recognition of intentionality behind actions

This comes from the system of plans that underlie each action. It is thus possible to infer the plan that motivated a given event given some known goal of the actor of that event;

(3) Filling in implicit information

Our representational system has a system of slots and default fillers of those slots. Thus, our system knows when it doesn't know something.

One problem that we have been particularly concerned with lately is the construction of an adequate representational system for conversation. Only some of the above mechanisms will be helpful in conversation. One of the things we need, and do not have at present, is an adequate model for the motivations of people when in interaction with other people. How much is appropriate to say when? How do you recognize the cues inherent in conversation that tell about what is appropriate in the conversation itself rather than those that just input event information? How do we represent such things?

HI.

Our system is quite good at inference, paraphrase, and translation into other languages. The last two are products of premises 1 and 2 in I. When the representation is language-free and unique, translation and paraphrase are facilitated.

The core of the language understanding problem is inference. The system of primitive actions helped our understanding of the inference problem as illustrated by the MARGIE system. Scripts solved the inference cutoff problem inherent in MARGIE and helped us to produce SAM. SAM is a reasonable model of story understanding and illustrates the power of CD/KS. The premise of predictive understanding and the predictive nature of CD/KS are what make SAM tick. Thus SAM has problems with unexpected inputs. One of the serious problems for us then, is the ability to shift to bottom up processing when predictions fail. This is something that PAM, a program that exploits goal and plan-based predictions, is intended to remedy. Still, there are probably times when even these will fail. 1 don't know how we will handle such instances, but I do know that human processors have a tough time with them also.

KNOWLEDGE REPRESENTATION SEMANTICS

Brian C. Smith

Within the Artificial Intelligence community there are numerous research projects involved in the construction of knowledge representation languages in addition to those represented here on this panel. Each is trying to provide a congenial formalism, whose structure will both aid in the process of initially representing knowledge within an AI system, and also help in understanding the structures that result. Although many of these efforts have met with a certain degree of success, none has taken firm hold. Instead we are confronted with a large number of complex systems, with different structures, dealing with different issues, which are difficult to understand and difficult to compare.

KRS-I (Smith. 1977). a formal system of Knowledge Representation Semantics, is an initial attempt to provide a coherent intellectual framework within which to understand systems of knowledge representation — a way to answer the question "what does all this mechanism mean?". The structure of KRS derives from a consideration of what the act of representation means within the context of building computational models of intelligence. Because it is being developed out of an interest not only in knowledge but also in active reasoning processes, the form of the semantic foundation that it provides differs from traditional theories of semantics. Instead of being interested in issues such as completeness, soundness, decidability, etc., KRS instead provides a formal structure in which to talk about such issues as memory chunking, locality of access, focus of description, abstractions, appropriateness of interpretive belief, etc.

Specifically, KRS rejects the following three assumptions, that have traditionally been held as axioms of any formal theory of meaning:

1. The idea that statements or expressions by themselves have meaning. Instead KRS formalizes the idea that "meaning" is something which makes sense only in terms of an active process interpreting a system of symbols.

2. The notion that "truth" is appropriate as a primitive semantic concept. This is not at all to say that the concept of truth is not important, but instead to reject its formalization as a binary and primitive notion, and also to reject the idea that deciding that truth of a sentence is the crucial aspect of uncovering its meaning. The truthfulness of a statement is instead seen to be a complex, subtle, contextual, and often useful description of that statement, which is neither primitive within the semantic theory, nor necessarily expressible in terms of the atomic symbols "TRUE" and "FALSE".

3. The assumption that it is in general possible or appropriate to say anything absolute or certain about the structure of the world being represented. What is considered to be important instead is what people or processes believe; KRS considers the only question that can be asked by a process to be what it believes, and also what • it believes that another process believes.

The overall framework of KRS-l is a formalization of 5 "levels'* or viewpoints from which to understand a symbolic description, or piece of representational structure:

1. A "message" level, which deals with the words or communication string on its own, without reference to the structure of the process that sent or received it

2. An "intermediate" level, embodying what is traditionally thought of as the syntactic structure of a message.

3. A "memory" level, formalizing notions of organization and accessibility.

4. A "belief" level, dealing with the active conclusions and inferences that an interpretive process will come to, based both on its previous beliefs and goals, and also on the structure of the memory level. This can be thought of as short term memory, although part of KRS-l is an account of how the issues dealt with at this level differ substantially from those of the memory level.

5. An "external" level, capturing the notion that a representation is a representation of something ~ this is the level which the interpretive process believes that the memory structures represent.

In addition to these levels, KRS also gives a precise account of "layers of meta-description" to characterize the relationship that holds between two descriptions when one describes the other. The .substance of KRS is a theory of the structure of, and the relationships between, these levels and layers, and of the role that an interpretive process can play within such a framework. (Note that "level" and "layer" are technical words naming two orthogonal dimensions of the semantic framework.)

KRS-l, the current version of the theory, deals only with the declarative structure of representational languages; although it identifies the place that an interpretive process must play in such a scheme (indeed formalizes the claim that you cannot understand the meaning of a symbol without understanding the processes that interpret that symbol), no attempt has been made to capture or describe the actual processing structures of an interpretive process. This is the direction towards which further work will be directed.

KRS and Natural Language Semantics There is no doubt that the structure of natural language provides significant evidence of the structure of human thought. However KRS is not specifically designed to be a theory of English semantics, for two reasons:

1. KRS-l is an attempt to be a theory of the semantics of computational knowledge representation languages, not of knowledge representation in the abstract. Hence the current objects of study are languages such as KRL, OWL, semantic nets, etc., rather than English

2. As opposed to the philosophy of OWL, there is no effort in KRS to account for specifically linguistic behaviour. For example, the goal of KRS is to make clean and precise all distinctions which seem cogent in identifying and solving subtle problems in terms of reasoning. Just because such a distinction is not apparent in the structure of English sentences will not be taken as any reason not to formalize the distinction.

KRS and a Description of the Interpretive Process:

One of the motivations for building a structured model of knowledge representation semantics is to provide a framework within which to describe the behaviour of an interpretive process. In traditional computer languages such as LISP, there is a well-defined and precise notion of evaluation which the interpreter is carefully designed to implement. However as we build more complex description systems, this strict model of evaluation begins to break down. For example, consider a system such as MACSYMA (Moses, 1974); one of the powers of that system is an ability to reason with symbolic descriptions (such as the "integral of X") without evaluating them.

KRS is designed to provide a model of an interpretive process; this model is the same as its characterization of any reasoning system. For example: suppose that an AI program is reasoning about a set of blocks on the top of a table. Suppose also that this program is "connected" to that table top by a video camera and robotic arm. As the program goes about its business, it builds up internal memory representations about the state of the world on top of the table, develops hypotheses about possible actions that it might take, explores what it thinks the consequences of potential actions would be, etc. Every so often, when it decides that it actually wants to do something, it reaches out and moves a block, or in some way changes the world about which it is reasoning.

One can draw a strong analogy between this program, and the interpreter that is running this program. In many ways their operations are very similar, except that where the domain of the blocks program is the table top, the domain of the interpreter is the program and representational structure of the blocks program. One can view it is a double-layered system, with the blocks program looking at the table and the interpreter looking at the blocks program. Where the blocks program reasons about a block partly by having a description of that block, so the interpreter can be seen to reason about the description of the block by having a description of of this description. In other words, the KRS characterization of intepretive process is of a reasoning system working at the meta layer.

KNOWLEDGE REPRESENTATION IN AIMDS AND ITS USE IN BELIEVER

N. S. Sridharan

Changing Worlds

Most representation systems have shied away from dealing with *updating* of information and have concentrated on reasoning with ..gjyen. collections of facts and general knowledge, The strength of the MDS and AIMDS systems is in having a systematic way of updating information. This allows one to adopt a "Hypothesize and Revise" paradigm in processing information in place of the more common search, methods that involve backtracking.

Premises

1. User suplies *general knowledge* about classes of objects (concrete objects, abstract objects, and spatial temporal logical relations among them). Knowledge about *changes* (actions, plans, hypothesis revision etc.) is supplied in the same formalism as the knowledge about the objects of change. *Exceptions* to the general rule are represented in a natural manner.

2. User supplies many knowledge sources *as though* they were independent; but the system *integrates* the knowledge as it needs them and uses them through a compiling process.

3. User defined processes are (a) defined on the *logical structure* of information and not on their syntactic structure; (b) defined so that they *receive* and *utilize feedback* from the knowledge sources.

4. Uniform system-defined procedures for asserting, querying, matching descriptions are available that permit user processes to have properties mentioned above.

5. Facts and general knowledge are clustered so that convenient pathways of control flow are established. Three levels of such chunking are available - called Frames, Dependency Networks and Contexts.

Englisb Dialog

The forte of the system developed primarily for BELIEVER, is in its ability to perform constructive information processing tasks (contrasted to deductive ones) using independently specified sets of constraints. The constraints in BELIEVER are in the form of internal consistency of the Person Model, of the plan structure, of the World Model and coherence between the plan structure and the two models. Presently there is no English input or output. In carrying out typed English dialog it is conceivable that a dialog model could be addilively specified ("added on") or knowledge about the domain of discourse be introduced to augment the general world knowledge. The hypothesize and revise paradigm may assist the dialog program in generating and maintaining only a limited number of alternative hypotheses and to avoid a backtracking structure of search.

Mechanisms

The important mechanisms in MMDS are

1. Definition of logical structure is made using a convention for introducing relation names along with typing of domain objects.
2. Semantic definitions of relations are given in terms of
   (a) properties of relations such as irreflexive, transitive.
   (b) expressions in a first order logical language whose vocabulary consists of user introduced relations their inverses, typed variables and constants.
3. Uniform procedures are available that use the semantic definitions for

   (a) forward inferencing (antecedent reasoning)
   (b) backward inferencing (consequent reasoning)
   (c) finding and automatically filling in information
   (d) providing a focus for updating information, and
   (e) checking semantic constraints and providing feedback to user processes.

4. User has some control over when and how these semantic definitions are used.

5. The system acquires through user interaction necessary information for hypothesis revision or updating. The collection of these acquired rules form the core of the feed back-driven user processes.

REFERENCES

Bobrow, Daniel G. and Winograd, Terry. "An overview of KRL, a knowledge representation language," *Cognitive Science,* 1:1 1977

Bobrow, Daniel C, Winograd, Terry, and the KRL Research Group. "Experience with KRL-0 -- One cycle of a knowledge representation language," *Proceedings of IJCAIV,* August 1977

Forgy, C, "A production system monitor for parallel computers," Technical Report, Carnegie-Mellon University, Department of Computer Science, Pittsburgh, Pa., 1977.

Forgy, C, & McDermott, J., "OPS, a domain-independent production system language," *Proceedings of JJCA1-V,* August 1977.

Grosz, B.J., "The Representation and Use of Focus in Dialogue Understanding," Dissertation, University of California, Berkeley, 1977.

Hawkinson, L.B., "The Representation of Concepts in OWL", *Proceedings of the Fourth IJCA1,* 1975.

Hendrix, G.G., "Expanding the Utility of Semantic Networks Through Partitioning," *Proceedings of the Fourth IJCAI;* "Representation of Semantic Knowledge and Semantic Aspects of Translation," in Walker, D.E. (Ed.), *Speech Understanding Research: Final Report, Project 4762,* Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, 1976, page 5.

Kay, M., "The MIND System," In Rustin (Ed.), *Natural Language Processing,* New York, Algonthmics Press, 1973.

Klahr, D. & Wallace, I., *Cognitive Development: An Information Processing View,* Erlbaum Associates, 1976.

Kowalski, Robert A., "Predicate Calculus as a programming language," *Proceedings of IFIP-74,* North Holland, 1974, pp 569-574.

McCarthy. John and Hayes. Pat. "Some philisophical problems from the standpoint of artificial intelligence," In Michie, D. and Meltzer. B. (eds), *Machine Intelligence 4.* Edinburgh, 1969. pp 463-502

McDermott, J., & Forgy, C, "Production system conflict resolution strategies," to be published in *Proceedings of Workshop on Pattern-Directed Inference Systems,* Honolulu, Hawaii, May 23-27, 1977.

McDermott, J., Newell, A., & Moore, J, "The efficiency of certain production system implementations", to be published in *Proceedings of Workshop on Pattern-Directed Inference Systems,* Honolulu, Hawaii, May 23-27. 1977.

Moses, Joel "MACSYMA - The Fifth Year," *S1CSAM Bulletin,* August 1974.

Newell, A., "Production systems: Models of control structures", in W.C. Chase (Ed.) *Visual Information Processing,* Academic Press, 1973, pp. 463-526.

Newell, A., & Simon, H.A., *Human Problem Solving,* Prentice-Hall. 1972.

Norman, D.A., and Rumelhart, D.E., and the LNR Research Group, *Explorations in Cognition,* San Francisco, California, W. H. Freeman and Co., 1975.

Quillian. M.R., "Semantic Memory," in Minsky (Ed.), *Semantic Information Processing,* Cambridge, Massachusetts, MIT Press, 1968.

Rieger, C. J., "Conceptual Memory: A Theory and Computer Program for Processing the Meaning Content of Natural Language Utterances," AI Memo 233, Artificial Intelligence Laboratory, Stanford University, Stanford, California, 1974.

Rychener. M., "Production Systems as a Programming Language for Artificial Intelligence Applications," PhD thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pa., 1976.

Rychener, M., & Newell, A., "An instructable production system: Basic design issues", to be published in the *Proceedings of a Workshop on Pattern-Directed Inference Systems,* Honolulu, Hawaii, May 23-27, 1977.

Schank, R.C., "Conceptualizations Underlying Natural Language," in Schank/Colby (Eds.), *Computer Models of Thought and Language,* Freeman, 1973, p. 197.

Schank, R.C., et al. *Conceptual Information Processing,* New York, American Elsevier, 1975.

Schmidt, C.F., and Sridharan, N.S., "Plan Recognition: A Hvpothesize and Revise Paradigm," to be presented at the IJCAI-V Conference, August. 1977.

Schubert, L.K., "Extending the Expressive Power of Networks," *Artificial Intelligence,* 1972, 2, 163-198.

Shapiro, S.C., "A Net Structure for Semantic Information Storage, Deduction and Retrieval, Advance Papers." Second International Joint Conference on Artificial Intelligence, London, 1971.

Simmons, R.F., "Semantic Networks: Their Computation and Use for Understanding English Sentences," In Schank/Colby (Eds.), *Computer Models of Thought and Language,* San Francisco, California. W. H. Freeman and Co., 1973.

Smith, Brian C. "A proposed system of knowledge representation semantics," Masters dissertation, Department of Electrical Engineering, MIT, June 1977.

Sridharan, N.S., "A Frame Based System for Reasoning About Actions," to be presented at the IJCAI-V Conference, August, 1977.

Sridharan, N.S., "The Frame and Focus Problems: Discussion in Relation, to the BELIEVER System," *A1SB Conference Proceedings,* Edinburgh, pp. 322-333.

Srinivasan, C.V., 'The Architecture of Coherent Information System: A General Problem Solving System," *IEEE Trans. Computers,* April 1976, (originally appeared in 3IJCAI).

Srinivasan, C.V., "The Model Space of the Meta Description System," Report SOSAP-TR-19, Department of Computer Science, Rutgers University, 1976.

Swartout, W.R., "A Digitalis Therapy Advisor with Explanations," MIT/LCS/TR-176 February, 1977, (See also the 1977 IJCAI Proceedings).

Szolovits, P., Hawkinson, L.B., Martin, .W.A., "An Overview of OWL, a Language for Knowledge Representation," MIT LCS Technical Memorandum, May 1977.

Woods, W.A., "What's in a Link: Foundations for Semantic Networks," in Bobrow and Collins (Eds.). *Representation and Understanding: Studies in Cognitive Science,* New York, Academic Press, 1975.