DIMENSIONS OF REPRESENTATION

BY DANIEL G. BOBROW

XEROX
PALO ALTO RESEARCH CENTER

DIMENSIONS OF REPRESENTATION

BY DANIEL G. BOBROW

CSL 75-5 JULY 1975

A set of questions is presented concerning representations of knowledge. The questions are organized in terms of a framework in which knowledge of a world-state is derived by mapping the world to a knowledge base. The dimensions of representation are defined in terms of design issues which must be faced or finessed in any representation. Issues considered include correspondence between operations on the world and on the knowledge base, organization of the mapping process, inference which would make explicit knowledge which would otherwise be implicit, philosophy and mechanisms of access to elements of the knowledge base, pattern matching in knowledge processing, types of self-awareness an understander system might have, and use of multiple representations. These dimensions are illustrated with examples from the literature. The differences between analogical and propositional representations are illuminated by consideration along the multiple dimensions of representation.

Key Words and Phrases:

artificial intelligence, representation, knowledge theory, understander systems

CR Categories:

3.6,3.36,3.42

XEROX

PALO ALTO RESEARCH CENTER

3333 COYOTE HILL ROAD / PALO ALTO / CALIFORNIA 94304

TABLE OF CONTENTS

I. Introduction....1
- A. Representation and mapping....1
- B. Three simple visual representations....4

II. Domain and Range....6
- A. Units and relations....6
- B. Exhaustiveness....7
- C. Verbal mediation....7

III. Operational Correspondence....10
- A. Updating and consistency....10
- B. History and planning....11
- C. Continuity....12
- D. Psychological modeling....13

IV. The Mapping Process....13
- A. Constraints on world states....13
- B. Procedural declarative tradeoffs....15

V. Inference....16
- A. Formal inference techniques....16
- B. Computational inference....18
- C. Meta-inferential techniques....19
- D. Preferred inferences....21

VI. Access....22
- A. Philosophy of association....22
- B. Access mechanisms....23

VII. Matching....25
- A. Uses for matching....25
- B. Forms of matching....26

VIII. Self-Awareness....28
- A. Knowledge about facts....28
- B. Knowledge about process....29

IX. Conclusion....29
- A. Multiple representations....29
- B. Analog representations....31

References....34

I. INTRODUCTION

Workers in cognitive science have worried about what people know, and how to represent such knowledge within a theory.[1] Psychologists such as Paivio (1974) and Pylyshyn (1973) have argued, for example, over two alternative forms for visual memory in humans. The style of their arguments, which we return to at the end of this paper, is to set up opposing characterizations and to argue about which one has more "natural" properties with respect to observed phenomena.

I claim that a more appropriate way of discussing the issues involved is to characterize each representation in terms of how it answers certain questions posed in this paper. I pose these questions in terms of a set of design issues one would face in designing or analyzing an *understander system*--a system (human or computer) which could use the knowledge to achieve some goal. I propose a framework for viewing the problems of representation. In this framework each of the design issues defines a dimension of representation--a relatively independent way of looking at representations.

In this paper I emphasize the structure of alternative solutions to the design issues. I illustrate the design options through three specific representations described here, and in examples from the literature. By considering representations along the separate dimensions, it often becomes apparent that a pair of seemingly disparate representations differ in very few significant features.

---

[1]In our recent book (Bobrow & Collins, 1975) we describe cognitive science as a new field containing elements from psychology, linguistics, computer science, philosophy, education, and artificial intelligence. This paper is a chapter in that book, as are a number of other papers referenced here.

A. Representation and Mapping

I propose here a framework where representations are viewed as the result of a selective mapping of aspects of the world. Suppose we take a "snapshot" of the world in a particular state at some instant in time. Call this state *world-state-1*. Through some mapping M, a representation (call it *knowledge-state-1*) is created which corresponds to world-state-1. This corresponds with world-state-1 in the sense that an understander has the alternative of answering questions about world-state-1 by directly observing the world state or by questioning the corresponding knowledge state (see Fig. 1). This implies, of course, the existence of a world-observation and knowledge-question function correspondence; simplicity of the mapping M, and simplicity of representing particular knowledge and questions must be considered in comparing representations of a world-state.

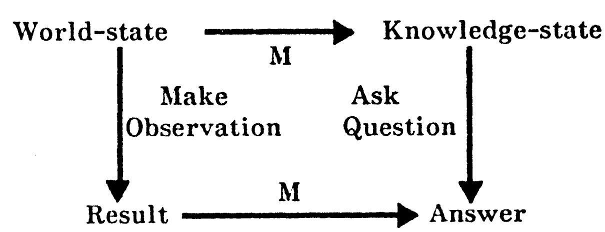

Fig. 1. Mapping between world and knowledge states. Answering questions should correspond to making observations and mapping the result.

The world at a particular instant is static, and all the facts about the world reflect a single consistent state. If we now augment our simple view, and allow *actions* which change some properties of the world, then we must have *model operations* which make corresponding changes in the knowledge state. For a model to be consistent, an updated world-state-2 must correspond to the updated knowledge-state-2 (see Fig. 2).

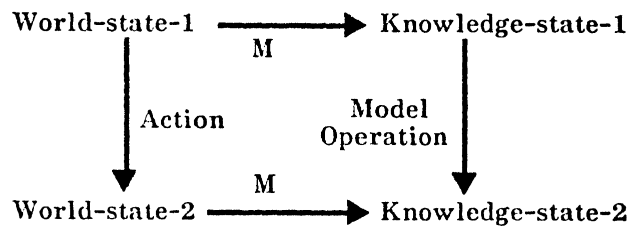

Fig. 2. A world-state can be changed by an action. An equivalent model operation should produce a change in the knowledge-state which corresponds to the changed world-state.

In terms of this simple framework for viewing representation, we can now look at a number of different design issues. I pose these as a series of questions to be asked about any mapping and the resulting representation of the world:[2]

Domain and Range: What is being represented? How do objects and relationships in the world correspond to units and relations in the model?

Operational Correspondence: In what ways do the operations in the representation correspond to actions in the world?

Process of Mapping: How can knowledge in the system be used in the process of mapping?

Inference: How can facts be added to the knowledge state without further input from the world?

Access: How are units and structures linked to provide access to appropriate facts?

---

[2]In constructing this list of questions, I have been influenced by the dimensional analysis used by Moore & Newell (1973) in describing their system MERLIN.

Matching: How are two structures compared for equality and similarity?

Self-awareness: What knowledge does a system have explicitly about its own structure and operation?

B. Three Simple Visual Representations

To illustrate some options concretely on certain dimensions, I use three different specific representations for the same simple domain--two-dimensional black and white scenes. I describe how each represents a visual scene which contains a square rotated so that one diagonal is horizontal.

Binary Matrix: Fig. 3a shows a two-dimensional binary matrix represention (MATRIX) of the spatial layout. A "1" is inserted in the matrix wherever the light intensity in the scene is below some threshold, and a "0" otherwise.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0
0	0	1	1	1	1	1	0	0	0
0	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

**Fig. 3a. A binary matrix visual representation.
A 1 indicates a light intensity below a certain level.**

A collection of connected 1s determines an object, with transitions between spaces containing 1s and 0s indicating the contours of an object.

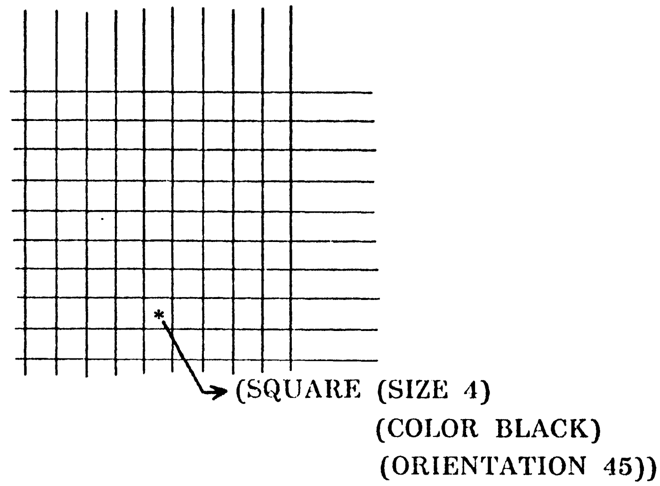

Fig. 3b. A Grid-positioned/feature oriented representation.

Grid-positioned feature: Fig. 3b shows what I call a grid-positioned feature representation (GRID) for a scene. An object is represented by a unit which specifies a set of features. The structure shown is of type SQUARE, with features specifying the size, color, and orientation of the square. The definition of SQUARE is not shown; it can be obtained given the type specification. The grid is used to locate objects in a scene. From a point on the grid corresponding to the location of the leftmost lowest point of the object, there is a link to the unit representing the object.

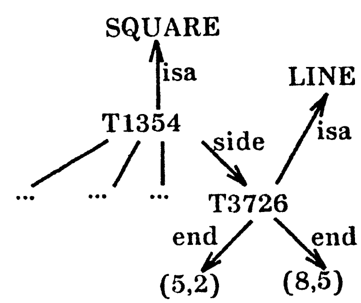

Fig. 3c. A semantic network representation.

Semantic Network: Fig. 3c shows a portion of a semantic network represention (NET) of the same visual scene. The units shown are a token of a square, tokens of sides of the square, and some number pairs representing the endpoints of the sides of the square. Only one of the endpoint sets are shown. Labelled links from one unit to another show the relations between the units.

II. DOMAIN AND RANGE

A. Units and Relations

The choice of units and structures reflects how one views the world one is modeling. A unit is something which can be used without knowing anything about its internal structure. This does not imply necessarily that it must not have any internal structure, just that there are occasions of use (e.g., inference rules) in which the existence of the unit is sufficient. In addition to its identity, response to a unit may be a function of its position in a larger containing structure, or special relation to other units, or to its internal form.

In choosing a representation for a particular world, some relationships can be stored explicitly and others need not be. For example, the size of the square is implicit in the matrix in MATRIX, as is the position of the square in GRID. These are reflections of what Hayes (1974) describes as the similarity between the medium of representation and the world, at least with respect to the relations being modeled.

Not all relations in a representation fully determine a portion of the world. For example, the relative position of two objects (A is left of B) may be implicit in locations represented in the model. Alternatively, this fact may be explicitly represented, with perhaps no absolute location information for either unit. How such "vague" predicates and partial information about the world are handled is an important characteristic of a representation. [Woods (1975) has a more complete discussion of problems of vague predicates.]

B. Exhaustiveness

A representation is exhaustive with respect to a property if for any object, if it has that property, that fact is stored explicitly. Not only does the model represent the truth, it represents the whole truth. Thus in an exhaustive representation of the objects present on the surface of a table top, any object not explicitly noted as on top is not there, and if no object is associated with a location, then that location is guaranteed to be empty. In an exhaustive representation all objects that exist are represented explicitly, and any universal proposition can be verified by testing all elements of this set of objects. Exhaustiveness is a second aspect of what Hayes (1974) refers to as similarity of structure of the medium.

One way for a visual representation to maintain the property of exhaustiveness is for the mapping to have the property of extracting a uniform degree of detail. An aerial photographer does map terrain this way whereas a cartographer may not. In the photo, it is guaranteed that no object within the field of view and larger than the resolution of the lens will be missing. The whim of the mapmaker determines the objects and features represented on a map. MATRIX is by nature exhaustive; GRID and NET can be made so by design. Human visual memory does not seem to have this property of uniform extraction of detail, or of exhaustiveness.

C. Verbal Mediation

Instead of mapping the world directly, people have constructed systems which map the world using natural language descriptions. There are many issues involved in building adequate representations of English language statements. Woods (1975) points, for example, to the subtle problems involved in representing relative clauses and verbal restrictions within a semantic network. In this paper, I focus only on the issues involving selection of basic units to represent linguistic information.

Word-Senses: An obvious choice for a unit is a single word. The relations chosen often are the case relations for verbs (Fillmore, 1968). This simplifies the mapping process by focusing on the obvious units and their grammatical relationships. A problem with this choice is that words are often ambiguous. Some systems finesse this issue by assuming that each word will have only one meaning within the domain of interest. Other systems face the issue by allowing individual words a number of different senses.

Several problems must be considered in systems which use word-senses. There is the obvious potential error in ignoring concepts for which there are no single words (or for which the user knows none), such as a single word to describe "those small orange cones used to divert traffic". A word-sense system must allow compound constructs to be used as well as atomic units.

Another problem arises if the system is forced to make an either/or decision, since use of the word may straddle two word-senses, even though word-senses in the dictionary are usually chosen so that they are distinguishably far apart. For example, consider the word "weigh" in "The butcher weighed the meat", which has a different sense than the same word in "The jury weighed the evidence." Instead of interpreting these sentences in terms of separate meanings for "weigh", we can consider its common core--"weigh" as "comparing an unknown with a contextually determined scale". If selection of a word-sense precludes using any part of any other meaning, then intermediate creative uses can be missed, particularly those arising in metaphorical use of language.

A system which uses word-senses must also provide a way of determining the equivalence of two different phrases. The sentences:

John sold the boat to Bill.

Bill bought the boat from John.

have identical factual meanings, although the difference in topicalization may be used to guide the storage of the information in long-term memory. Recognition of the equivalence of these paraphrases for purposes of inference requires translation rules to convert from one form into another, or separate inference rules for each form [see Woods (1975), and Simmons (1973) for a more complete discussion of this issue].

Semantic Primitives: If simplification of the paraphrase problem is made a focus of the representation design, the mapping can be designed to translate all input to a canonical form, so that identity of meaning is equivalent to identity of form. One way to do this is to expand all input to expressions involving only a small fixed set of primitive units as the basic relations. Schank (1973) gives a number of arguments for use of such an expansion, and describes a set of eleven ACTs (Schank's primitive units) which are useful for this purpose. All actions in his system are expressed in terms of this primitive set, and the predicates define states which result from these actions. Schank claims that this representation has the additional advantage that inferences which should be made when a new sentence comes in can be keyed to the individual ACTs rather than having to be stored with each word or word-sense.

As an argument against expansion to primitives, one notes that there are significant inferences which must be made on the basis of particular combinations of ACTs and states for a particular word. Consider the differences in obvious inferences between these two statements:

> *John thought something, which caused him to do something, which caused a male actor to become in a state of worst possible health.*
>
> *John killed him for a reason.*

This is only a mild caricature of expansion to primitives. The point is, to recognize the situation from the longer paraphrase, a more complex match is required than for the more compact one. Thus there is a tradeoff between the types of operations that can be done easily in the two representations. This is really a form of the tradeoff of processing at input time versus processing at time of use of information.

III. OPERATIONAL CORRESPONDENCE

One issue of major concern in representation is the correspondence between action and structures in the physical world, and operations and representational forms in a model. Simple actions in the domain should be reflected in simple operations in the representation.

A. Updating and Consistency

A major design problem in modeling actions is updating the representation with respect to a chain of changes caused by a single action. For example, suppose at time t1 a cup of coffee is on a table. At time t2 the cup is pushed over the edge of the table, and at time t3 all has settled down again after the actions starting at t2. It is easy to see how a system might represent world-state-1. It is easy to see how the action at time t2 might be represented. The problem comes in determining how a representation might reflect the facts that if a cup falls off a table, the cup has changed position, but the table has not and the contents of the cup (the coffee) is no longer either on the table or in the cup. How does the system determine which of the facts true at t1 are true at t3? Must every model operation have associated with it a way of checking for all possible implications? Or should relations and objects in the world be represented by active entities which check for conditions affecting them? Small systems have been built on each assumption, and the tradeoffs are just beginning to be explored.

A complementary problem exists if not all operations in the model correspond directly to actions in the world; a unitary operation may correspond to only one element of an action. The representation may then allow knowledge-states to be constructed which cannot be realized in the world. For example, in GRID, the extent of each object is not indicated on the grid. Therefore, if in the knowledge state an object is moved so that its attachment point is adjacent to another object's attachment point, there is no obvious violation, but the objects may overlap unacceptably in the real world.

B. History and Planning

In the simple mapping of static world-states, time is represented only implicitly in terms of changes in the world, and corresponding changes in the knowledge base. A more sophisticated representation would allow the simultaneous representation of two different world-states, so that a history can be stored. A major problem is how to find and represent shared pieces of the two knowledge states. Problems related to shared structure have been discussed extensively in the artificial intelligence literature using the phrase "the frame problem", for example by Raphael (1971). Unfortunately, this use of the word "frame" differs in meaning from the current use of the term "frame", as, for example, in Minsky (1975) and Winograd (1975).

In the previous section on updating and consistency I assumed the need for only one set of "true" facts, true for the time the world-state was modeled. If a history is to be stored, then those facts which *were* true but are not now, must be distinguished. One alternative is to associate with every fact time bounds for the truth of that fact. Sometimes this is difficult in that the start or end times may not be known. An alternative proposed by Sandewall (1971) is to mark changes as they occur. A fact is assumed true at any time t2 if it was true at some earlier time, t1, and the system cannot prove that the fact has become untrue. Thus facts need not be repeated for each instant, but it may be costly to check on the truth of a fact.

A related problem occurs in planning. Planning is a search for a series of actions to bring about a particular desired world-state. In conducting the search, shared knowledge and updating problems must be dealt with. In planning, the changes are not real; they result from modeling activity, not world activity. A search can be made by actually carrying out a plausible sequence of operations and testing the result. Exploring the search space of operations is a difficult task, and needs to be guided by common sense.

In order to allow backup in case of error, a copy of the original world state must be kept, or provision must be made to allow "undoing" operations (reversing their effects in the model). In addition, if two worlds are to be compared, there must be some way to save information about alternative hypothetical states of the world. An obvious possibility is to have multiple worlds in which operations are applied to separate copies of the knowledge or modeling base. Here the system must provide some way of knowing which information is shared and which is unshared.

C. Continuity

As we have defined them, operations change the knowledge-state to reflect the differences in the world-state at the beginning and endpoints of an action. One argument for certain types of models concerns the intermediate states the model should go through when certain operations are performed. Model operations can be implemented so as to permit only small incremental changes in the model.

For example, in any of the visual representations, one could imagine that a large change of location by an object could only be made as a sequence of small changes in the representation. Thus in MATRIX, rotations and translations might be made by moving one cell of the matrix one unit at a time. In GRID, rotation might add only a small increment to the orientation value, perhaps limited by the resolution with which such information is specified. In either case, if one wished to model a reasonably large movement, the object representation would have to proceed incrementally through space from the initial position to the desired position, perhaps checking in the model for interpenetration of objects on the path. If operations had enforced continuity, other constraints on the trajectory could also be checked, as well as possible transient effects such as blocking of light.

D. Psychological Modeling

If a purpose of the representation is to provide a psychological model of some mental activity, then some correspondence must be defined between measurable resources used by a person and invocation of some operations in the model. If the obvious choices were made, then in MATRIX the time for mental rotation of an object would be proportional to the area of the object (number of 1s in the matrix). In NET, it would be proportional to the number of endpoints of lines; and for GRID it would be independent of the shape and size.

Imposing continuity on an operation like rotation allows the assumption that the change in the representation takes time proportional to distance traveled. This, of course, is analogous to transformations in the real world, despite the fact that the underlying format of the representational model might not normally be considered analogous to the world. Models with this property for humans seem to be implied by the data of Cooper & Shepard (1973). In their experiments, subjects were asked to compare a rotated figure with a possibly identical one in a standard position. The time to make the comparison was a linear function of the amount of rotation, but independent of the complexity of the line drawings used (Cooper, 1975).

IV. THE MAPPING PROCESS

A. Constraints on World States

In our discussion thus far it has been assumed that units and relations correspond to particular objects and relationships in the world. If knowledge in the model is to help in the mapping, some facts must represent constraints over sets of world states (perhaps even all mappable states). In order to do this a system can use formulas or structures containing *variables*. One can interpret such a structure as follows: if appropriate constants are substituted for the variables in the structure, then the relationship expressed will be true.

Mapping design issues then center around mechanisms for specifying and finding constants which satisfy the appropriate constraints; this is related to the issue of determining the size and structure of units retrieved and used in the mapping process.

Restrictions on Variables: In addition to satisfying the relations indicated in formulas, variables are often subject to other restrictions. A simple restriction is on the type of the entity which can be substituted for a variable. For example, the value of an attribute COLOR can be specified to be one of the color-names. In some cases, restriction on a range of values is also specified; for example, a day of the month must be an integer and between 1 and 31. Restrictions on variables can be used to check possible substitutions, or can be used to help in a search for objects in the world which are appropriate.

Other restrictions, aside from range, can arise from interaction of selection of variable substitutions. In one reading of the statement "every professor X loves some student Y", then the choice of Y is dictated by the identity of the element which is chosen for X. This is typical of issues which arise in quantification. It is here that formal representations based on the predicate calculus have advantages, in that the subtleties of the connections have been worked out. Woods (1975) gives examples of a number of problems in quantification as expressed in English, and works out forms of representation for quantification for use in a semantic network.

Higher Level Structures and Mapping: Predicate calculus and semantic network representations tend to impose only a local organization on the world. Much of the thrust of recent research (see Bobrow & Collins, 1975) deals with organizing information at higher structural levels. Such higher-level structures help in the description and instantiation of structures as complicated as birthday parties or a complete story. Default values are often provided for variables in such structures so that a priori guesses can be used to provide a complete picture with little or no processing. Such guesses would most often be right (a person usually has two legs), and need only be verified by a quick test. Kuipers (1975) gives an example using a frame for a standard clock, in which an object identified as a clock is represented as having hands, although this was assumed from default, and in the example was an incorrect assumption.

Larger structures provide a conceptual framework on which to hang inputs. By forcing inputs to fit within an expected framework, however, the system will see only what it wants to. Alternatively, using the data to drive processing will cause extensive search as numerous low-level combinations of units will be found which cannot be used in larger structures. Bobrow & Norman (1975) discuss tradeoffs between such concept-driven and data-driven processing, suggesting both are needed, and must interact.

B. Procedural Declarative Tradeoffs

Information used in processing can be isolated in declarative data structures or can be embedded in procedures for achieving special purposes. In the extreme, one can take the view that "knowing" is "knowing how to" and that all behavior is engraved in programs. Hewitt (1971) has been a major proponent of expressing all knowledge as procedures. The issues between declarative and procedural underlying structures are extensively discussed by Winograd (1975), so I only summarize the arguments here. Declarative languages provide economy of representation (many uses for the same knowledge), and human understandability and communicability. They rely on general procedures using special problem-dependent data. Procedural embedding emphasizes use of specific procedures for specific problems, allowing easy use of control information (second order knowledge) and easy representation of process-oriented information. Winograd characterizes these features in terms of modularity versus interaction, and makes a preliminary proposal for joining declarative and procedural representations.

V. INFERENCE

Not all of the facts in any knowledge-state need be kept explicitly in a representation. If partial knowledge is available in the system, then some set of explicit facts may have implications, that is, determine further facts which satisfy the constraints represented in the particular knowledge state. *Inference* is the process of deriving implicit facts from the initial set of explicit formulas according to some fixed rules of inference without interaction with the outside world. The form of inferences available and the structure of data to support these inferences are important design decisions for a system.

I distinguish among three different forms of inference. The term *formal inference* covers the family of techniques used in predicate calculus representational systems. The term *computational inference* describes a process in which facts are derived through bounded known computation. The term *meta-inference* covers techniques by which knowledge about the structure and content of the data base is used to derive further facts consistent with the original set. A major forcing function in design of a representation can be the desire to make a particular set of inferences easy. These *preferred inferences* are often what give a representation much of its power.

A. Formal Inference Techniques

In formal systems, facts about the world are represented in quantified formulas. To infer a new explicit fact one produces a formal proof, with a chain of intermediate formulas produced through modus ponens, resolution, or another standard syntactic method. All the facts which support an inference are thus available for inspection, in contrast to computational inference discussed in the next section.

A formal basis for inference has a number of strong implications for the representation chosen. McCarthy & Hayes (1969) have pressed a strong case for using a formal (predicate calculus) substrate for an understanding system. Advantages cited include the use of theorem proving techniques which are not domain-dependent. Thus these techniques are applicable whether the information represented concerns trip planning, children's stories, or the physical world of robots. Formal systems often have the property of completeness; that is, the proof techniques guarantee that if a fact *can* be proved from those available to the system, then, given enough time, that fact *will* be proved to be true. The logic of quantifiers and their interaction has been extensively worked out, and the predicate calculus takes advantage of this long history of careful thought.

The other side of each of these arguments is as follows: If only general theorem proving is used, then special facts about the domain (for example, classifications of facts as useful in particular types of inferences) are made difficult or impossible to use. The property of completeness is often not really useful because the condition "given enough time and space" is often unfulfillable. Moreover the system is built on the assumption that it contains only a consistent set of facts. If what is being represented are the beliefs of an individual at some time, then this set of beliefs may indeed not be consistent, or at least not expressed in a consistent manner. For example, the generalization that "all birds can fly" can usefully live in a system which contains the specific facts "ostriches cannot fly" and "an ostrich is a bird". In classical logic, the existence of these three inconsistent facts would allow the deduction of any arbitrary fact.

Two techniques are often used to work around the problem of contradictory beliefs in the same data base. An ordering principle can be used, where specific facts are considered before generalizations. Alternatively, the quantifier "all" can be reinterpreted to mean "all, unless I tell you about an exception". In either case, formal properties of the quantifiers disappear, and standard proof procedures can no longer be used.

B. Computational Inference

In some systems a specialized procedure is used for computing certain facts about the state of the world. For example, for a world consisting of a set of objects placed on a two-dimensional plane, there may be a list of that set of objects and their x-y coordinates. Additional facts can be derived using *procedural specialists*, such as a "left-of" specialist that uses the coordinate information to answer the question about whether the object X is to the left of object Y. In SOPHIE (Brown & Burton, 1975), procedural specialists use the contingent voltage table and a resistance-connection table for the circuit to answer questions about currents and power dissipation in any element in the circuit.

There are two points to make to distinguish this form of inference from the formal techniques. First, control information--which facts to use next and how--is built into the procedure. Therefore search procedures are not included in this class of computations. Although this is often efficient, it can lead to some rigidity in how the procedure works and which parameters it can use. The idea is that, by specialization to a particular assumed environment, special case data and control tests can be avoided.

Second, no intermediate results are available as in formal inference; no justification of any result is given. For example, the input to the circuit simulator of Brown & Burton are values of circuit elements. A relaxation method--make an approximation, find the errors, try again--determines a consistent set of voltages across elements. The result is accepted as valid without proof; the only guarantee of correctness is at the level of initially proving the program correct (or just debugging it).

This works if all inputs are valid, and no anomalous cases occur. If errors occur, a simple procedural model can only throw up its hands. One alternative, used in PLANNER (Hewitt, 1971), is to have a set of specialists for doing individual tasks; if one specialist fails, the system reverts to its earlier state (backtracks) and tries another specialist. In PLANNER, desired inferences are classified by their syntactic form and content. Procedures to make such inferences are invoked on the basis of the form of the fact to be proved. All inferences in PLANNER are carried out by procedural specialists written by the user.

Sussman (1973) has built a program for debugging programs which is based on computational inference. The procedural steps in the program to be debugged are augmented by a set of "intention statements" which can be checked against the program for various known forms of "bugs" (errors in programming). Winograd (1975) describes a system with a procedural base, which also contains a declarative description of procedures that are invoked. If the compiled version (the one which is unitary and leaves no trace) runs into unexpected problems, then the task can be rerun in a more careful mode using the procedure description.

C. Meta-Inferential Techniques

Some systems have been designed to find facts which are not necessarily derivable in a formal way from the set already present, but which are consistent with such a set and may be useful.

Inductive Inference: One class of techniques, inductive inference, uses a set of facts to form the basis for a general rule for expressing relations. The general rule is consistent with the given data but may not necessarily be correct; it may be later contradicted by additional data. Using a general rule to replace specific data can save space in information storage. It may also provide a basis for new theorems in the system. Brown (1973) discusses a number of problems in building an inductive inference system which occur even in a very limited world; one of the worst problems is the existence of faulty data.

Inference by Analogy: Another class of inference techniques goes under the general rubric of inference by analogy. In inference by analogy, if certain criteria of similarity are met between two situations, then a result that pertains to the first situation can be assumed to pertain to the second situation. Collins, Warnock, Aiello, & Miller (1975), working on a tutoring program called SCHOLAR, discuss a particular form of inference by analogy which they call *functional inference.* They distinguish major conditions they call *functional determinants* which are critical in allowing a geographical location to have a particular property. For example, the latitude and altitude of a place are the major functional determinants of the type of climate at that place. SCHOLAR uses the following rule for inference by analogy.

> If a property P has functional determinants F and G, and F and G are identical for place 1 and place 2, then barring information to the contrary, if place 1 has property P, then assume place 2 has that property as well.

Thus since Los Angeles and Sydney, Australia are both at sea level and at 33 degrees latitude, their climates should be similar.

Learning this type of functional knowledge is an important part of human learning in general, and such functional rules allow one to generate many reasonable answers without formally sufficient data. In general, for an analogic inference, the criterial properties of a situation with respect to some result must be marked and stored in the representation.

Self Knowledge Inferences: Another class of meta-inferences taken from the SCHOLAR program is based on the system's knowledge of its own internal structure. SCHOLAR uses information about the importance of particular properties, and level of relevance of facts it has about a particular place. This extension of the exhaustiveness property discussed earlier allows determination of negative answers based on not finding information in the data base; without such knowledge the system would often be forced to reply "I don't know".

As a simple example, consider how the SCHOLAR program answers the question, "Is oil a major product of Chile?" SCHOLAR knows that copper is a major product of Chile, and oil is a major product of Venezuela. It also knows facts which are less important than the major products of Chile, so it assumes it knows all products that are of major importance. It thus responds, "No, oil is probably not a major product of Chile."

As another example, consider a question discussed by Norman (1973), "What is Charles Dickens' phone number?" Most humans (and hopefully most intelligent systems) will be able to answer "I don't know" immediately without having to do a long search. Again this is based on knowing what is known, and how easily accessible such knowledge is.[3] Norman proposes multiple stages of search, with an initial filtering done on the basis of knowledge of the system's own knowledge.

D. Preferred Inferences

Each system has certain inferences which can be made more easily than others. Often this is designed into a system. For example, in most semantic nets a preferred inference attributes to an individual any property of the general class to which it belongs. For example, Fido would inherit all the properties of a generic dog, e.g., he has four legs, he barks, etc. These preferred inferences often give a system much of its power; this has certainly been true with semantic networks (Quillian, 1969).

Some systems derive preferred inferences on the basis of examples. That is, for a particular set of inputs, they derive an example which satisfies the inputs, and then check to see whether a suggested subgoal is true in the example. From general specification of a geometric figure, Gelernter's (1960) geometry machine drew (inside the computer) an example of a figure satisfying the specification. Any constructions and any hypotheses considered

[3] A Chas. Dickens is listed in the Palo Alto phone book; I looked it up after Allen Newell asked me how I knew there wouldn't be such a listing (as he found out) in the Pittsburgh directory. If a person knew a Charles Dickens other than the well-known author, this fact would probably be unusual enough to make it immediately accessible when such a question were asked; thus, with this assumption the question could again be answered with assurance.

were first checked against this particular instance to see if they were reasonable.

VI. ACCESS

Use of the appropriate piece of knowledge at the right time is the essence of *intelligent* mental operations. Two different issues arise in the consideration of access to data and procedures. The first concerns the philosophy of which elements to link together. The second concerns mechanisms which are used for access.

Since access and storage are inverse operations, there is a tradeoff between work done at each time. I assume that retrieval (access) is done significantly more often than storage; therefore I focus only on the access issue, and assume for this exposition that any necessary work has been done to allow the access regimes discussed.

A. Philosophy of Association

In each system there are implicit access links between elements of a single structure, and links which join structures. The former reflects which things in the world are viewed as unitary structures. The latter is used to facilitate internal processing such as making inferences. In predicate calculus representations, the natural structure is the formula. A single formula contains a number of different relations among units. The relation is the critical item, and so organizational aspects of the structure are based around selection of the relations. It is made easy to determine which relations have been used together, and harder to find all the potential properties of an individual, and the relations in which it has appeared.

In semantic networks, the organizational aspects of objects are emphasized, and the relations appear primarily in the interconnection between the units. A semantic network makes immediately accessible the kinds of relations that an individual participates in. It is possible to test whether information about a known individual is new or redundant, inconsistent or derivable from previous relations. Use of variables, and constraints between variables are harder to represent.

In more direct models, such as GRID or MATRIX, access is usually defined in terms of spatial location. Near neighbors of a point are directly accessible, and properties of that point are easily available; for example, the contents of that point can be found immediately.

One way of building structures larger than single formulas is to consider contexts in which relations are used. If a particular set of facts, or network structures, are used for understanding a particular situation, then that entire context can usefully be retrieved at one time. Those contexts themselves may be organized into still higher-level contexts. For example, the meaning of the words "cost" and "buyer" may best be understood in the context of knowledge about commercial transactions; further implications will come from a context generally applicable to a monetary economic system.

Another possible organizational structure is in terms of scenarios. Here, some higher-level structure which one wants to impose on the world is used to tie together otherwise disparate facts. The problem of putting together the individual structures of a representation in terms of higher-order structures has been discussed by Schank (1975), who places causal links between propositions in a paragraph; and Rumelhart (1975), who proposes a structure which describes well-formed stories. Winograd (1975) deals with the issue of associating specialized procedures with frames. The frame not only acts as an organizational structure for data, but for procedures as well.

B. Access Mechanisms

Suppose a unit of information is placed in a known location, for example starting at some address A in a computer. Then this information is directly accessible (without search) to a process which knows A. That is, there is an operation basic to the computer which retrieves the contents of a cell given the address of that cell. The address A can be stored as an element in another unit of information (call it B). If B contains A, we say that B has a pointer to A. This is how semantic networks represent links between elements. One of the main features of semantic networks is this explicit representation of interconnections between memory units.

If direct access is not available, then a *retrieval* mechanism must be invoked which will take a description of a desired unit, and search memory for a unit which fits the description. Ordering and structuring memory can speed up a retrieval search. The cost is paid at storage time, either in placement of items or in updating indexes.

It is important to note that usually only part of a description (a "key") is used in the search, and then the potential candidate is *matched* against the full description to determine its appropriateness. Sometimes, although a single direct pointer is not given, a list specifying a set of possibilities is provided. Then a description is used only for checking the possible candidates.

The description of a unit to be accessed can be constructed from both stored and dynamic information. For example, current context can delimit the set of possible elements which are of interest, and only a brief description need be used to discriminate one of these. Pronominal reference in English makes use of such assumed context for successful operation. Bobrow & Norman (1975) make a case for context-dependent descriptions being the primary basis for access in an intelligent system.

An access mechanism which is much discussed, but which has not yet been used in any artificial intelligence systems, is an active content addressable memory. In such a memory, a description of a desired unit would be broadcast to many (perhaps all) active memory units. Each would compare its own contents with the request, and answer if a good enough match were found. Problems which must be faced include specification of how good the match must be, how to get the information back to the requester, how to deal with conflicts, and how to resolve timing problems if more than one request is active at a time. Because of present hardware and software limitations, such a system has not been tried, although procedural systems such as CONNIVER (Sussman, 1972), have used software to simulate some of the properties. A goal pattern for a procedure is specified, and an access mechanism invokes procedures which have been stored with a "trigger" pattern which matches the goal pattern. Bobrow & Raphael (1974) describe this pattern-directed invocation, and a number of other properties of the new artificial intelligence programming languages.

Another access mechanism which can use an active memory system is the "intersection" technique simulated in many semantic network models. Here access is specified in terms of two key elements which are both to be associated by a chain of direct links with the desired item. From each of these keys the network is searched by following pointers from each key, in a breadth first fashion, until an element is reached by search from one key which has previously been reached from the other. More than one intersection may be found if parallel active search is going on. Models using this type of access have been proposed for human processing. Collins & Quillian (1972) among others have conducted a number of interesting experiments which give some evidence for this type of search in human language processing.

VII. MATCHING

A. Uses for Matching

Matching as an operation can be used for a number of purposes within an intelligent system: *classification, confirmation, decomposition* and *correction.* To determine the identity of an unknown input, a number of possible labeled patterns can be matched against the unknown. The unknown can be *classified* in terms of the pattern it matches best. This is the paradigm for simple pattern recognition. In retrieval, a possible candidate to fit a description may be *confirmed* by the match procedure. If it matches well enough, the retrieval and match together provide a pattern-directed access capability. A pattern with substructure can be matched against a structured unknown, and the unknown *decomposed* into subparts corresponding to those in the pattern. A parsing system is a complicated pattern matcher whose purpose is to find substructures corresponding to patterns in the grammar. In certain matches, what is critical is the form or direction of the error in the match. In hill climbing or relaxation techniques, a first approximation to a solution is *corrected* by use of this error term. Kuipers (1975) discusses using errors of prediction as a guide in pattern recognition.

B. Forms of Matching

Systems frequently use matching for one or more of the above purposes, and purpose can be confounded with the form of matching done in the system; we describe three basic forms of matching, syntactic, parametric, and semantic, and a mode of forced matching.

Syntactic Matches: In syntactic matching, the form of one unit is compared with the form of another, and the two forms must be identical. In a slight generalization, a unit may have variables, which can match any constant in the other. Further complications involve putting restrictions on the types of constants a variable can match. A common use of syntactic matching procedures is to find appropriate substructures by matching variables which fit into parts of larger structures. Another step in the generalization is to allow the pattern matcher to be recursive, so that the matcher is called to determine if a subpiece of a pattern matches a subpiece of a unit. Bobrow & Raphael (1974) describe classes of variable restrictions and pattern matching in current AI programming languages.

Parametric Matches: In syntactic match, a binary decision is made. A pattern either does or does not match. In a parametric match, a parameter specifies the goodness of any match. In such a match, certain features of a pattern may be considered essential, others typical and hence probably should be there, and others just desirable in an element to be matched. A goodness parameter can account for how many of which features can be found. Ripps, Shoben & Smith (1974) hypothesize that people use a parametric match using levels of feature comparison. For example, they claim a person would classify a particular picture of an animal as a bird if sufficient features presented in that picture match those of a "typical" bird.

Semantic Matches: In a semantic match, the form of elements is not specified. The function of each element in the structure is specified; then the system must engage in a problem-solving process to find elements which can serve that function. For example, a table could be specified to be a horizontal surface on top of a support which keeps the surface at a height of about 30 inches. This does not at all specify the form of the support, which could be anything from a box to a cantilever from a wall. This type of specification, separating form from function, seems necessary to allow the flexible definitions that humans seem capable of handling.

Forced Matches: Moore & Newell (1974) in their MERLIN system, discuss a mapping process in which one structure is viewed as though it were another. Matches of corresponding items in the structures are forced if necessary. Forcing such matches allows certain operations applicable to one unit to be used in conjunction with the other. For example, if you were in a locked room and wished to get out, you could break open the window if you had a hammer. If no hammer were available, it might occur to you to view your shoe as a hammer; the sole would be forced to match the handle of the hammer, and the heel the head. Bobrow & Norman (1975) discuss procedures for building in the generalizability required by such forced matches by using minimal descriptions. These descriptions serve to aid in the identification of relevant matches, and to handle the necessary applications of the constraints on these variables.

VIII. SELF-AWARENESS

An important dimension of system design is whether the system has explicit knowledge of its own workings. This dimension has not been well explored in representation systems, and so I give here only a menu of different kinds of self-awareness that might be built into a system.

A. Knowledge about Facts

Exhaustiveness of a representation with respect to a property is a form of self-knowledge which we discussed with respect to operational correspondence and meta-inference. It is generalizable to the level of relevance, as in the SCHOLAR system. A related property is a level of importance or interest associated with classes of facts. This type of knowledge is useful in forward inferencing schemes in which resources have to be allocated; inferences based on interesting or important new facts should be made first.

Criteriality is a term used to describe the relevance of the identity or truth of some element in a match. Becker (1973) uses the adjustment of criterialities as a basis for automatic generalization of experience. Another class of knowledge about facts concerns the belief status of a fact. Values of belief between true and false can be used, as well as the basis on which the belief was acquired. For example, a system may remember that it was told a particular fact by Richard, and therefore it is much less likely to be true. Providing criteriality or expected degree of validity of information is important when a contradiction is encountered. This is the type of knowledge that a system must have in order for it to be able to correct errors in its own procedures. Other useful facts about facts are characterizations of situations in which they are useful. Classification of the kinds of facts known, and the importance for different functions is a level of self awareness whose utility we describe in the section on inference.

B. Knowledge about Process

In modeling interactions with the outside world, the system needs to predict its own capabilities to plan a strategy in which information gathering cannot all be done before starting an action sequence. For example, in planning a route, it must be able to realize that at a certain intersection it will be able to look for a street sign.

Other process-knowledge information is relevant to a system which has different strategies for solving problems with special characteristics. Characterization of a problem should be a first step in deciding when to apply domain-specific heuristics. Information useful for scheduling competing processes is important in multigoal systems. Such knowledge includes resource requirements for procedures, and a priori and dynamic estimates of success of particular problem solving routines. I believe all of these levels of self-awareness will be necessary for us to build intelligent understander systems.

IX. CONCLUSION

A. Multiple Representations

It is often convenient (and sometimes necessary) to use several different representations within a single system. In this way, it is sometimes possible to combine the advantages of different representational forms within one system. The use of multiple representations leads to two primary problems: choice and consistency.

Choice: In a system with multiple representations, a particular fact may be represented in several ways. In such cases, a system must contain a mechanism to choose which form to use for any particular fact. For example, the location of an object may be given by its coordinates, or in terms of its relative location to some other object, or it may be placed in a grid with the location implicit in the grid point on which it is located. If different representations are used, mechanisms are needed to transform information in one representation to that of another. Sometimes, however, the information in one representation does not allow a reasonable transformation to another. For example, knowing that A is to the left of B does not position A precisely enough to allow it to be placed on a grid, even if B is on that grid.

Representations determine the ease of answering certain questions, and of performing updating operations. At times, it is best to enter information directly into one representational form and then, from there, compute how to enter it into the other form. Thus an object's position might first be entered by its coordinates, and then its position relative to all others computed and inserted into the appropriate representation. Questions about its relative position and its absolute position then can be answered with equal facility.

Consistency: In a system with multiple representations the same information can be stored in more than one form. When one form changes, the other forms must be checked for consistency. For example, if the left-right relations of an object have been stored, and the object is moved, all those relations must be recomputed. An alternative is to maintain a primary data representation, such as the positional information. Secondary information can be represented in procedural form, with special procedures to compute the desired results quickly.

Updating is a more serious problem in representations in which facts may have been inferred on the basis of a large number of other facts. The multiple representation problem compounds the problems of single representation consistency, updating, and planning I discussed in Section III.

Efficiency: Major considerations in use of multiple representations are tradeoffs between computation and storage, and availability of special techniques for achieving efficiency; for a particular process all information may be transformed to the preferred representation. For example, Brown & Burton (1975) use a dual representation system for electronic circuits. Circuit calculations are performed by a circuit simulator which provides descriptions of particular, consistent states of the circuit. The simulator implicitly embeds in its operations knowledge of the interactions and feedback among circuit elements. A semantic network in their system, which stores propositional information, is excellent for answering many types of questions; but it would founder on the feedback issue.

B. Analog representations

In psychology, a current debate rages over how visual information is represented in human memory--whether or not it is stored in "analog" form. For example, Sloman (1971) points to implicit interaction as an important argument for analogic representations. Pylyshyn (1973) argues that if information is stored as images it must have a uniform degree of detail in the representation. Given the known fine detail a person can sometimes store, uniform extraction implies an overload of information in picture memory. Paivio (1974) rejects the uniform-detail position in arguing for images. He claims, however, that propositional models can not have appropriate continuity in operations, thus failing to model the Cooper-Shepard results I described in IIID. This characterization over-simplifies the arguments, but indicates the dimensional nature of the disagreements.

Having representational dichotomies such as *analogical* versus *propositional* requires, I think, that we make overly sharp distinctions. In this paper, I illustrated the properties inherent in a choice of representation for visual scenes by discussing three possibilities: MATRIX, NET, and GRID. The units of MATRIX are only the visual elements, and relative location is an implicit relation between two units. NET has named symbols as units, and named relations linking them, with no implicit relations. GRID has two types of units, grid points to record positions, and symbolic units represented in a list of property value pairs at some of the grid points. Only MATRIX is "naturally" exhaustive, though the other two can be made so explicitly. Whereas MATRIX seems "obviously" analogical, and NET propositional, it is harder to decide about GRID. I believe that such debate is best viewed by considering claims made along separate dimensions.

The most distinguishing feature of these representations is along the dimension of access. Properties of a point are directly accessible from the location in MATRIX. In NET such information can only be found by search and computation. Access to a unit as an entity is direct in GRID and NET, and requires a search in MATRIX. In GRID, but not in NET, one can access a square (or any unit) directly knowing its center of mass. In NET, but not GRID, the coordinates of the corners are a directly accessible property of an object. I did not define properties of MATRIX, GRID, and NET with respect to operational correspondence, mapping process, inference, matching or self-awareness. Often in an isolated model, significant differences in theory rest on which dimensions of representation are *not* considered.

From these and other examples we can see that there are representations which can be considered more or less analogical with respect to different properties. If the same relationships implicit in the representation are implicit in the world (such as betweenness), or if continuous operations in the world have continuous analogs in the representation (such as movement of objects), then, for those relationships and operations, the representation may be considered more analogical. Sharp categorization (such as a fixed set of size descriptors) and formal operations (such as modus ponens) make a representation seem more propositional. A single representation may be characterized as propositional in some parts, and analogical in others.

This paper provides the reader with a set of questions to ask about representations of knowledge. The questions are organized in terms of a mapping framework, with dimensions corresponding to design issues which must be faced or finessed in any representation. It is my experience that viewing representations along these multiple dimensions allows more complete and coherent evaluations and comparisons.

ACKNOWLEDGMENTS

I would like to thank Don Norman without whose help and encouragement this paper would never have been written. I am also grateful to the following people for comments which greatly improved the final result: Rusty Bobrow, Lynn Cooper, Ron Kaplan, Martin Kay, Allen Newell, and Terry Winograd.

REFERENCES

Becker, J. D. A model for the encoding of experiential information. In R. C. Schank & K. M. Colby (Eds.), *Computer models of thought and language.* San Francisco, Ca.: Freeman, 1973.

Bobrow, D. G., & Collins, A. Representation and Understanding: Studies in Cognitive Science. San Francisco, Ca.: Academic Press, 1975.

Bobrow, D. G., & Norman, D. A. Some principles of memory-schemata. In D. Bobrow & A. Collins (Eds.) *Representation and understanding: Studies in cognitive science.* San Francisco, Ca.: Academic Press, 1975.

Bobrow, D. G., & Raphael, B. New programming languages for artificial intelligence research. *Computing Surveys*, 1974, *6(3)*, 153-174.

Brown, J. S. Steps toward automatic theory formation. *Third International Joint Conference on Artificial Intelligence.* Stanford University, August 1973, 121-129.

Brown, J. S., & Burton, R. R. Multiple representations of knowledge for tutorial reasoning. In D. Bobrow & A. Collins (Eds.) *Representation and understanding: Studies in cognitive science.* San Francisco, Ca.: Academic Press, 1975.

Collins, A. M., & Quillian, M. R. How to make a language user. In E. Tulving and W. Donaldson (Eds.), *Organization of memory.* New York: Academic Press, 1972.

Collins, A., Warnock, E. H., Aiello, N., & Miller, M. Reasoning from incomplete knowledge. In D. Bobrow & A. Collins (Eds.) *Representation and understanding: Studies in cognitive science.* San Francisco, Ca.: Academic Press, 1975.

Cooper, L. A. Mental rotation of random two-dimensional shapes, *Cognitive Psychology*, 1975, 1, 20-43.

Cooper, L. A., & Shepard, R. N. Chronometric studies of the rotation of mental images. In W. G. Chase (Ed.), *Visual information processing.* New York: Academic Press, 1973.

Fillmore, C. J. The case for case. In Bach & Harms (Eds.), *Universals in linguistic theory.* Chicago, Ill.: Holt, 1968.

Gelernter, H. Realization of a geometry theorem proving machine. *Proceedings of 1959 International Conference on Information Processing.*, 1960, 273-282.

Hayes, P. Some problems and non-problems in representation theory. *Proceedings of the A.I.S.B Summer Conference, Sussex University*, 1974, 63-79.

Hewitt, C. Description and theoretical analysis (using schemata) of PLANNER: A language for proving theorems and manipulating models in a robot. Ph.D. Thesis (June 1971) (Reprinted in AI-TR-258 MIT-AI Laboratory, April 1972.)

Kuipers, B. A frame for frames: Representing knowledge for recognition. In D. Bobrow & A. Collins (Eds.) *Representation and understanding: Studies in cognitive science.* San Francisco, Ca.: Academic Press, 1975.

McCarthy, J., & Hayes, P. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer and Michie (Eds.), *Machine intelligence 4.* Edinburgh University Press, 1969.

Moore, J., & Newell, A. How can MERLIN understand?. In Gregg (Ed.), *Knowledge and cognition.* Baltimore, Md.: Lawrence Erlbaum Associates, 1973.

Norman, D. A. Memory, knowledge, and the answering of questions. In R.L. Solso (Ed.), *Contemporary issues in cognitive psychology: The Loyola symposium.* Washington, DC: Winston, 1973.

Norman, D. A., & Bobrow, D. G. On data-limited and resource-limited processes. *Cognitive Psychology*, 1975, *7*, 44-64.

Norman, D. A., Rumelhart, D. E., & the LNR Research Group. *Explorations in cognition.* San Francisco, Ca.: Freeman, 1975.

Paivio, A., Images, propositions, and knowledge. Research Bulletin No. 309. London, Canada: Department of Psychology, The University of Western Ontario, 1974.

Pylyshyn, Z. W. What the mind's eye tells the mind's brain: a critique of mental imagery. *Psychological Bulletin*, 1973, *80*, 1-24.

Quillian, M. R. The teachable language comprehender: A simulation program and theory of language. *Communications of the ACM*, 1969, *12*, 459-476.

Raphael, B. The frame problem in problem-solving systems. In *AI and heuristic programming*. Edinburgh University Press, 1971.

Rips, L. J., Shoben, E. J., & Smith, E. Structure and process in semantic memory: A featural model for semantic decisions. *Psychological Review*, 1974, *81(3)*, 214-241.

Rumelhart, D. E. Notes on a schema for stories. In D. Bobrow & A. Collins (Eds.) *Representation and understanding: Studies in cognitive science*. San Francisco, Ca.: Academic Press, 1975.

Sandewall, E. Representing natural language information in predicate calculus. *Machine Intelligence 6*. Edinburgh University Press, 1971.

Schank, R. C. The structure of episodes in memory. In D. Bobrow & A. Collins (Eds.) *Representation and understanding: Studies in cognitive science*. San Francisco, Ca.: Academic Press, 1975.

Simmons, R. F. Semantic networks: Their computation and use for understanding English sentences. In R. C. Schank & K. M. Colby (Eds.), *Computer models of thought and language*. San Francisco, Ca.: Freeman, 1973.

Sloman, A. Interactions between philosophy and artificial intelligence. *Artificial Intelligence 2*, 1971.

Sussman, G. J. A computational model of skill acquisition. MIT-AI Laboratory AI TR-297 (August 1973).

Sussman, G., & McDermott, D. From PLANNER to CONNIVER - A genetic approach. Fall Joint Computer Conference. Montvale, N. J.: AFIPS Press, 1972.

Winograd, T. Frame representations and the procedural-declarative controversy. In D. Bobrow & A. Collins (Eds.) *Representation and Understanding*. San Francisco, Ca.: Academic Press, 1975.

Woods, W. A. What's in a link: Foundations for semantic networks. In D. Bobrow & A. Collins *Representation and understanding: Studies in cognitive science*. San Francisco, Ca.: Academic Press, 1975.